أساسيات قواعد البيانات

تأليف

مارك جيلينسون

تعريب

د . مهندس/سرور علي ابراهيم سرور
استاذ مشارك بقسم الأساليب الكمية
كلية الاقتصاد والإدارة
جامعة الملك سعود ـ القصيم

مراجعة

الدكتور/بو علام محمد بن جيلالي
أستاذ مشارك بقسم الاقتصاد (سابقاً)
كلية الاقتصاد والإدارة
جامعة الملك سعود ـ القصيم

تقديم

دكتور / عبد الله بن عبد الله العبيد
عميد كلية الاقتصاد والإدارة
جامعة الملك سعود ـ القصيم

دار المريخ للنشر
ص. ب: ١٠٧٢٠ ـ الرياض: ١١٤٤٣ ـ تلكس ٤٠٣١٢٩
المملكة العربية السعودية ـ تلفون ٤٦٥٨٥٢٣ ـ ٤٦٤٧٥٣١

أساسيات
قواعد البيانات

حقوق النشر :

DATABASE Step-by-Step, by: Mark L. Gillenson

الطبعة العربية

بسم الله الرحمن الرحيم

تقديم

بسم الله الرحمن الرحيم والصلاة والسلام على أشرف المرسلين سيّدنا محمد وعلى آله وصحبه ومن تبعه بإحسان إلى يوم الدين.

وبعد

يدخل هذا الكتاب عن قواعد البيانات في إطار سلسلة كتب الكمبيوتر/التي تقوم كليّة الاقتصاد والإدارة بترجمتها ونشرها باللغة العربيّة في إطار سعيها لخدمة القاريء العربي. وقد اختير هذا الكتاب خصيصاً ليقدّم أساسيّات قواعد البيانات حيث أنه لا يحتاج إلى خلفيّة كبيرة في تشغيل البيانات كمتطلّب سابق مع افتراضه بأن القاريء يفهم الأساسيّات الأوليّة للكمبيوتر ومفاهيم برمجته الأوليّة.

وقد قام الدكتور/ سرور علي ابراهيم سرور الأستاذ المشارك بقسم الأساليب الكميّة بكليّة الاقتصاد والإدارة بترجمته وراجعه الدكتور بوعلام بن جيلالي الأستاذ المشارك بقسم الاقتصاد بكليّة الاقتصاد والإدارة. ونوجّه إليهما وافر الشكر على ما قاما به من مجهود في إعداد هذا الكتاب أثابهما الله خيراً على عملهما هذا.

ونرجو أن يكون هذا الكتاب إضافة علميّة جديدة للمكتبة العربيّة.

د. عبد الله بن عبد الله العبيد
عميد كلية الاقتصاد والإدارة
جامعة الملك سعود
فرع القصيم

مقدمة المترجم

«قل لو كان البحر مداداً لكلمات ربي لنفد البحر قبل أن تنفد كلمات ربي ولو جئنا بمثله مدداً» صدق الله العظيم.

هناك ضرورة لتقديم هذا الكتاب عن قواعد البيانات للقاريء العربي. وتنبع هذه الضرورة من سماع الكثير عن نظام قاعدة البيانات المعروف باسم dBase III plus. وقد سمع المترجم من الكثيرين أن هذا النظام هو نظام قاعدة بيانات بينما أن النظام هو نظام إدارة قواعد بيانات. وهناك اختلاف كبير بين قاعدة البيانات وإدارة قاعدة البيانات. وتظهر هذه الاختلافات واضحة جليّة في هذا الكتاب.

كما أن الكتاب يقدّم أساسيّات قواعد البيانات وتكوينات قواعد البيانات مع الدخول في تفاصيل مفيدة للقاريء. وكما ذكر مؤلّف الكتاب فالقاريء الذي في حاجة إلى فهم أساسي لقواعد البيانات فقط سيكون هذا الكتاب هو ما يبغيه فعلاً. أما القاريء الذي يتوقّع أن يتوسّع في معرفته لقواعد البيانات فيقدّم له هذا الكتاب قاعدة صلبة تسمح له بالاستمرار في تفاصيل أكثر لقواعد البيانات ونظم إداراتها المختلفة مميّزاً أنه هناك اختلافات بين قواعد البيانات ونظم إداراتها.

ولا يسعنا إلا أن نقدّم خالص الشكر لسعادة الدكتور/ عبد الله بن عبد الله العبيد عميد كليّة الاقتصاد والإدارة بجامعة الملك سعود لتشجيعه الدائم على نشر المعرفة باللغة العربيّة الخاصة بأهم مجال من مجال حياتنا وهو الكمبيوتر باللغة العربيّة.

كما لا يفوتنا أن نشكر كل من الأستاذ محمد محمد محمد رضوان والأستاذ عبد المنعم علي غزالة على ما بذلاه من جهد. ونرجو من الله أن نكون قد وفّقنا الى تقديم كتاب جديد للمكتبة العربيّة.

وعلى الله قصد السبيل.

المترجم

المحتويات

مقدمة

قاعدة البيانات - دراسة تخزين وإدارة البيانات - هي برعم لحقل في الحقول سريعة الانتشار من حقول علم الكمبيوتر وتشغيل البيانات. وكما في نشأة أي فرع من فروع العلم فقد بدأت كدراسة لجزء بسيط من المعرفة قام بها مجموعة محدودة من العاملين في الصناعة والعاملين في الجامعات وطلبة الدراسات العليا بالجامعات. واليوم نجد أن المواضيع الجزئية لقواعد البيانات قد ازداد كل من عددها وعمقها زيادة هائلة. وقد اتجه الكثير من العاملين المختلفين في تشغيل البيانات وغير تشغيل البيانات والمنظمات الصناعية وغير الصناعية الى هذا الموضوع وكأحد المواضيع الجامعية فإنه يتجه بسرعة الى مقررات علم الكمبيوتر وإدارة الأعمال للطلبة الدارسين للحصول على درجة الليسانس أو البكالوريوس.

وهناك خاصية سيئة أو مدهشة لمجال قواعد البيانات وهي أنه لكي تفهم قواعد البيانات جيداً أي لحذف الغموض الخاص بكيفية تحقيق ذلك بدقة للعمل في الأجزاء المعدنية وأجزاء السيليكون الباردة من الآلات فيجب تمثيل كم هائل من المعلومات المتداخلة. ومن الصعب فهم سبب استخدام تكوين بيانات هرمي وذلك بدون فهم مشاكل تكرار البيانات والعلاقات المتعددة. ومن الصعب فهم كيفية كسب اتصال بسجلات في تكوين شبكي وذلك بدون فهم أساسيات طرق الاتصال ومن الصعب فهم الحاجة إلى وظيفة إدارة البيانات بدون فهم أساسيات منهج قاعدة البيانات. مثل هذه القائمة من المفاهيم المتداخلة يمكن أن تستمر إلى مالانهاية في هذا المجال.

والغرض من هذا الكتاب هو توضيح أساسيات قواعد البيانات والمدى الواسع للمواضيع الجزئية لقواعد البيانات بطريقة لا تترك شيئاً للخيال. وفي الناحية الأخرى فإنه لا يحاول أن يغطي كل تفاصيل هذا المجال تماماً. وهو لا

يحتـاج إلى خلفية كبـيرة في تشغيل البيـانـات كمتـطلب سـابق ولكنـه يفـترض أن القاريء يفهم أساسيات الكمبيوتر الأولية والمفاهيم الأولية لبرمجة الكمبيـوتر فقط. ويعـالج الكتـاب الموضـوع بطريقـة تقدميـة تؤكد الثقـة من فهم موضـوع قواعـد البيانات.

وهذا الكتاب يهـدف لاستخدامـه ككتاب أسـاسي لأول مقرر من مقـررات قواعد البيانات أو كمقدمة لهذا المجال لمهني تشغيل البيانات أو لغيرهم ممـا يتطلب عملهم معرفة بيئـة قواعـد البيانـات ويشمل الكتـاب مواد عن تعـريفات أسـاسية للبيانات وكذلك تكويناتها الأساسيـة وعن طرق الاتصـال وخواص ومنـاهج نـظم إدارة قواعد البيانات وتصميم قواعد البيانات والمفـاهيم الإدارية للبيئـة ومواضيـع أخرى مرتبطة بالمجال.

أخيـراً أود أن أترك الفكـرة التـاليـة مـع علماء الكمبيـوتـر مـع دفـاعي عن إحساسهم المشترك الخاص بالـدقة. في محـاولة هـذا الكتاب الـدخول في تفـاصيل مذهلة لقواعد البيانات وتركيزه على المفاهيم الأساسيـة أثناء حصره لمحتـوى الحقل فقد يعمم هذا الكتاب أو يبسط أكثر من اللازم بعض النقاط.

وأنا أقترح ببساطة أن العائد منه يستحق. فالقـراء الذين في حـاجة إلى فهم أساسي لقواعـد البيانـات فقط سـوف يتمتعون من هـذا العائـد بينما القـراء الذين يتوقعون أن يستمروا في قواعد البيانات بعد أن يقوموا ببناء قاعـدة صلبة بقـراءتهم هذا الكتاب سيكون أمامهم الكثير من الفرص بالاستمرار في تفاصيل أكثر مع دقة أكبر في مقررات أخرى تالية أو في الخبرة العمليـة أو في قراءتهم الكتب الأخـرى. ويميـل هذا الكتـاب أن يكون مقـدمة صـديقة للمستفيـد لمحتويـات حقل قـواعد البيانات. ووجهات النظر المعـروضة في هـذا الكتاب لا تعـرض إلا وجهات نـظر مؤلفة فقط.

الفصل الأول
مقدمــة

مقدمة Introduction

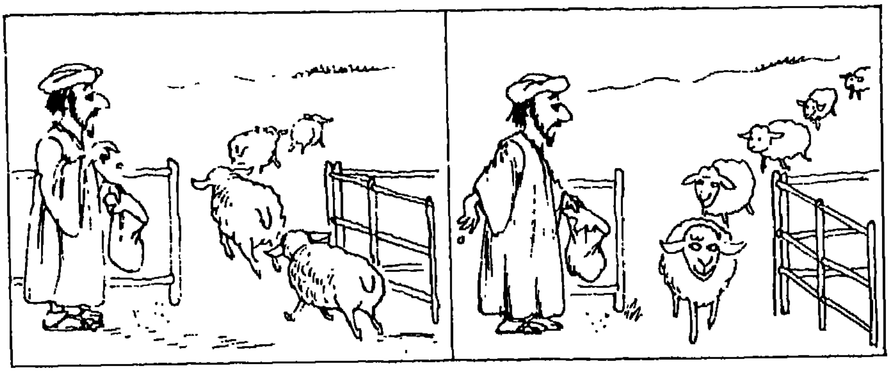
هـذا الكتاب عن البيـانات. الحقـائق. قطع من المعلومـات مطلوبـة لإدارة أعمالنا وحكوماتنا وبنوكنا وجامعاتنا وكـل نوع من أنـواع المحاولات البشريـة. فهو عن طـرق إدارة البيانـات نفسهـا بحيث أن البيـانـات التي نحتـاج إليهـا بـالضبط نجدها عند الحاجة لها فهـو عن أجهزة الكمبيـوتر. والـطريقة التي تعـرف بها بيئـة استخدام الكمبيوتر بأنها قاعدة بيـانات هي حـل للمشاكـل التي يتسبب فيها الكم الهـائل من البيـانات التي نخـزنها فيها. إنـه عن الجزء الـرئيسي ربما أهم جـزء من أجـزاء تقنية تشغيـل البيانـات في وقتنا الحـالي فهو عن كـل فرد يصـاحب تشغيـل البيانات ويكون في حاجة إلى فهم أساس.

دعنا نبدأ بتتبع اهتمام الإنسان بالبيانات منذ بدء الخليقة حتى يومنا هذا.

البيانات وأجهزة الكمبيوتر Data and Computer

نشأة حفظ السجلات

لقد اهتم الإنسان بـالبيانـات منذ 12000 سنـة على الأقـل. وبينما نصـاحب اليوم مفهوم البيانات بالكمبيوتر فقد كـان هناك طـرق بدائيـة لمعالجـة البيانـات في التاريخ: وفي واقع الأمر مـا زالت بعض هذه الـطرق مستخدمـة حالياً أيضاً. في العـراق حاليـاً يمكن مشاهـدة الـرعـاة يتتبعـون قـطعـانِ الغنم عن طـريق عـدهم باستخدام الحصى شكل 1.1. كلما خرج أحد أفراد القطيع للمرعى يضع الـراعي حصاه في كيس صغير موجود بيده. وعندمـا يعود أحـد أفراد القـطيع الىٰ الحـظيرة يلقى الراعي بإحدى الحصوات من الكيس وبهذا يكون لدى الراعي طريقه لمعرفة

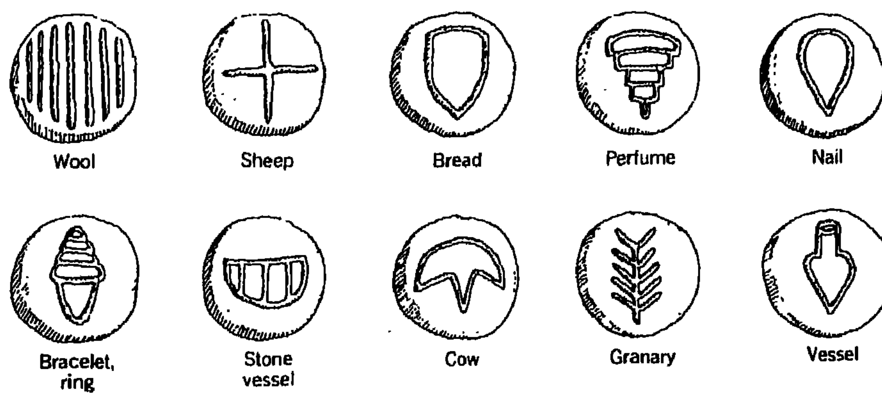


شكل 1.1
استخدام الراعي لكيس الحصى في تتبع قطيع الغنم

ما إذا كان القطيع كاملاً أم لا. وفي واقـع الأمر فـإن طريقـة حفظ السجلات هي صيغة من أقدم الصيغ التي عرفت.

الكشف عن الآثـار في منطقـة زاجـروس Zagros في إيـران والتي تعـود إلى 8500 سنة قبل الميلاد أظهر أنها كان بها عـدّادات طينيه والتي استخـدمت في حفظ السجلات في صور بدائية للمحاسبة. مثـل هذه العـدّادات وجدت في مـواقع من تركيا الحالية إلى باكستان الحـالية وكـذلك في الخـرطوم وتعـود إلى 7000 سنة قبـل الميلاد.

ومنذ 3000 سنة قبل الميلاد وفي مدينة سوسة Susa الإيرانية وصل استخدام هذه العدّادات أعلى درجة تعقيد. فهذه العدادات يطبـع عليها عـلامات خـاصة، شكل 1.2، ثم تختم على أقراص فخارية مجوفة والتي كانت تمثـل موتـورات شحن



شكل 1.2 أقراص فخارية كانت تستخدم في تسجيل المعاملات التجارية

تصـاحب السلع التجـاريـة في نقلهـا. وقـد كـانت تمثـل العـدادات كميـة السلع المنقولة. ومن الواضح أنه لم يكن العبث بها يتم دون كسر الأقراص الفخارية التي طبعت عليها هذه العدادات. لقد كانت النقوش الموجودة خارج القرص الفخاري والأختام المشمولة تقدم سجلات أكثر. وكانت النقوش الخارجية تشمـل كلمات أو مفاهيم مثل (منقول) أو (مودع) أو (مرحل).

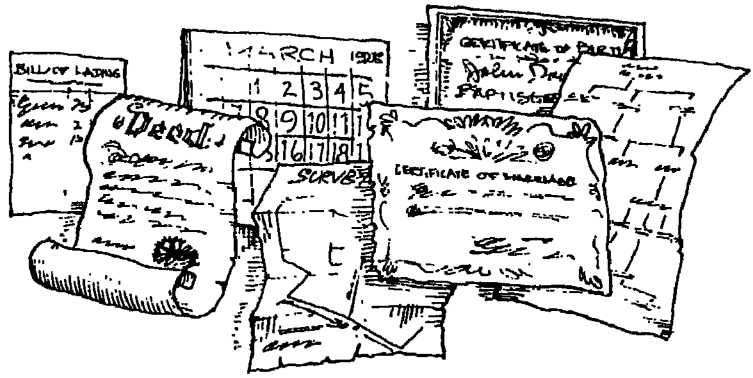
وفي نفس وقت وجود حضارة سـوسة تقـريباً حفظ النـاس في مدينـة أوروك Uruk في سـومـاريـا Sumaria السجـلات في نصـوص فخـاريـة. فقـد وصفـوا باستخدام الصور المرسومة والأعـداد الرمـزية مبيعـات الأرض والعمليات الجـارية للأعمال بما في ذلك الخبز والمشروبات والأغنام والحيوانات والملابس.

هناك طرق قديمة أخرى لحفظ السجلات تشمل تخزين العدادات على هيئـة قطع ونتوءات في أجزاء خشبية وعلى هيئة عقـد في الحبال. وقـد استمر استخـدام الطريقة الأولى في انجلترا حتى فـترة القرون الـوسطى كـما استخدم هنـود امريكـا الجنوبية الطريقة الأخيرة.

البيانات عبر العصور

يمكن إرجـاع الاهتمام الأولي بـالبيانـات الى وجود المـدن فحضارات الصيـد البسيطة والجمع وكـذلك الحضـارات الزراعيـة لم يكن لها سـوى استخدام محـدود لمفهوم البيانات. ولكن مع رؤيتنا لمدينة سوسـة ومدينـة أوروكا وللحضـارة الحاليـة التي بدأت مع عصر البرونز Bronze فقد تغيرت طريقة الحياة تغييراً لا رجعة فيه. فبـدايات الانتـاج الغزيـر وتخصص العمالـة واستخدام النقـود وخدمـات المقايضـة ومنتجات الفرد اللازمة للحياة كل ذلك يتطلب حفظ بيانات في سجلات.

ومـع مرور الـوقت حفظت بيـانات مختلفـة وبيانـات أكـثر. ويشمـل ذلـك تقويمات وبيانات تعداد سكان وبيـانات اجـراءات حصر وسجلات ملكيـة أراضي وسجلات زواج وبيانات عن مساهمات المعابد وأشجار العائلات شكل 1.3. وعلى التجـار أن يحفظوا تتبعهم للمخـزون والشحنات والأجـور بالإضـافـة إلى الانتـاج نفسه. ومع ذهاب الزراعة خلف مستوى المادة وتطورهـا الى مرحلة الإقـطاع كان هناك حاجة إلى حفظ بيانات عن الكمية المنتجة للاستهلاك وللمقايضة والتي تحفظ كبذور للسنة القادمة.



شكل 1.3
زيادة عدد أنواع البيانات مع تطور الحضارة

لقد وجد الصليبيون Crusade منذ القرن الحادي عشر الميـلادي حتى أواخر القرن الثالث عشر الميلادي. واحد الآثار الجانبية لهم هو النظرة الواسعة للعالم من وجهة نظر الأوروبيين مع تزايد الرغبة في التجارة. وإحدى طرق التجارة الشـائعة في ذلك الوقت كانت عن طريق عمل مشاركة بين التجـار وقباطنـة السفن والملاك وذلك لتسهيل الرحلات التجارية. وقد أحضر هذا المستوى المتزايد من التعقيد في التجارة معه جولة أخرى من حفظ السجلات مرتفعة التعقيد وبصفة خـاصة حفظ السجلات ذات المدخلين double - entry book keeping.

لقـد نشأ حفظ السجـلات ذو المدْخلين في مـراكز التجـارة في القرن الـرابع عشر بإيطاليـا. ومن أوائل الأمثلة المعـروفة مثـال عن تجارة في جنسوه يعود إلى عـام 1340 م. وقد انتشر استخدامه تدريجياً إلا أنه في عـام 1494 م. (بعد حـوالي 25 سنة من استخدام أول مطبعة متحركة في فينيسـا). قام راهب Fransiscan Mank يـدعَى لـوقـا بـاسيـولي Luca Pacioli بنشر كتـابـة عن «Sum de Arithmetica, Geometrica, propertioniet et proportionatita» وقـد كان لهـذا العمل تـأثير هام في انتشار استخدام حفظ السجلات ذات المدخلين. وكعنصر منفصل بالطبـع فإن زيادة استخدام الورق والطباعة ساهم في زيادة تقدم حفظ السجلات أيضاً.

ومع قلة شمول التجـارة الإيطاليـة أصبحت الدول الأخـرى أكثر نشـاطاً في التجـارة وفي الاستمرار مـع البيـانـات وحفظ السجـلات وأكـثر من ذلـك، فمـع الإقلال من المشاركة التجارية الوقتية واستقرار المنظمات التجارية التي شيدت فقـد

أصبح من الضروري وجود أنواع أخرى من البيانات مثال ذلك تقارير الأرباح والخسائر السنوية على عكس التقارير التي كانت تصاحب كل مغامرة. وفي عام 1673 م تطلب قانون التجارة «Code of Commerce» في فرنسا أن يعد تقرير موازنة كل سنتين بالنسبة لكل رجل أعمال.

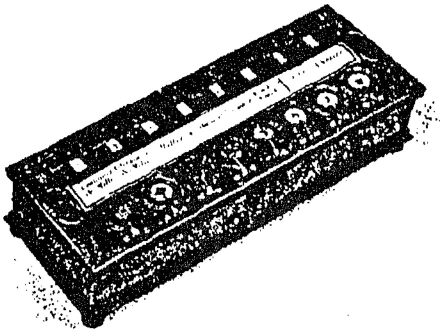
وحدات عمل الحسابات الأولية

لقد بدأ الناس يهتمون خلال القرن السابع عشر الميلادي كذلك بمعدات اجراء الحسابات التي يمكن أن تقوم بتشغيل بياناتهم آلياً حتى إذا كان ذلك بطريقة بدائية. وإحدى المعدات الأولى والمعروفة جيداً أنتجها بليز بسكال Blaise Pascal في فرنسا في العقد الرابع من القرن السابع عشر، أي 1640 م. وقد أعدها لمساعدة والده في أداء عمله كمحصل ضرائب شكل 1.4. وقد كانت الآلة عبارة عن صندوق صغير يحتوي على تروس متصلة ببعضها البعض وكانت قادرة على عمل عمليات جمع وطرح. وفي واقع الأمر فإن هذه الآلة هي أساس عدادات السرعة في السيارات المستخدمة حالياً.

وحوالي عام 1694 م بنى جوتفريد وليلم ليبينز Gottfried Wilhelm Von Leibniz آله أكثر تعقيداً تحتوي على اسطوانات وعجلات لها أسنان متداخلة مع بعضها شكل 1.5. وقد كانت الآلة قادرة على إجراء عمليات ضرب وعمليات قسمة مع إدخال عدد به عدة أرقام في البداية وتمثيل العدد الآخر عن طريق إدارة أحد التروس عدد من المرات. وقد عانت آلة ليبنز في الاستخدام العملي من المهارات الآلية الفقيرة التي كانت موجودة في وقت بسكال إلا أن أساسياتها استخدمت في آلات الحسابات الميكانيكية المعاصرة.

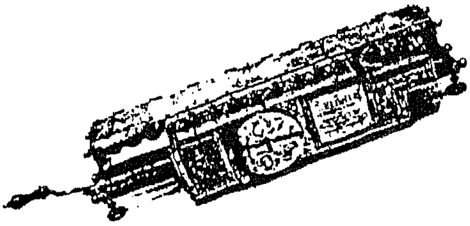
وقد اخترج يوسف ماري جاكورد Joseph Morie Jacquard الفرنسي في عام 1805 م آلة تنتج نمطاً بطريقة تلقائية وذلك أثناء عملية نسج الأقمشة. وقد كان قلب الآلة عبارة عن سلسلة من البطاقات الموجودة بها ثقوب وتسمح الثقوب بنسج المواد بتسلسل معين بحيث ينتج النمط المطلوب شكل 1.6. وبينما كان نول نسيج جاكورد ليس آلة للحسابات إلا أن طريقته في تخزين الأنماط كثقوب في بطاقات مثقبة كان وسيلة ذكية جداً لتخزين البيانات Data Storage والتي كان لها أهمية كبيرة بالنسبة لآلات اجراء الحسابات التي تلت ذلك





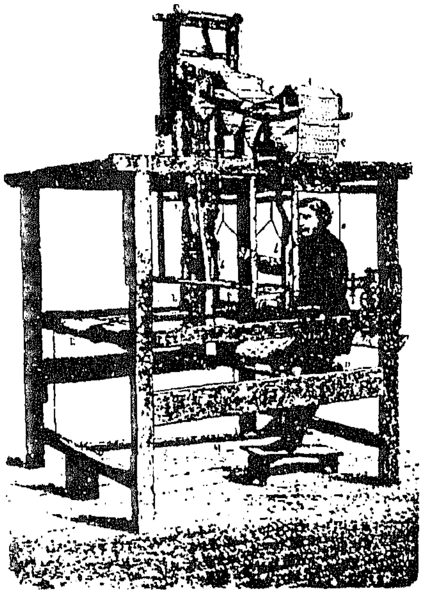
شكل 1.4
بسكال وآلة الجمع التي انتجها
(بإذن من أرشيف IBM)





شكل 1.5
ليبنز وآلة الضرب التي شيدها
(بإذن من أرشيف IBM)

وآخر أذكياء اجراء الحسابات الأوليين هو في واقع الأمر رجل سابق لعصره وهو شارلز باباج التاجر الإنجليزي الذي ظهر في القرن التاسع عشر. فأثناء العشرينيات والثلاثينيات من هذا القرن 1820 م و 1830 م طور باباج وبني آلة سماها آله الفرق شكل 1.7. وقد كانت الآلة معقدة بالنسبة لوقت ظهورها وكانت قادرة على عمل جداول لسلسلة من اعداد مستخلصة من كثيرات حدود معقدة. وكانت هذه الجداول مطلوبة في الملاحة والفلك وقد بدأ باباج عام 1833 م في التفكير في اختراع آخر سماه آله التحليل. ولم يتم هذا العمل (مرة أخرى كانت



شكل 1.6
نول نسيج جاكورد
(بإذن من أرشيف IBM)

حالة التجارة متخلفة بالنسبة لهذا الاختراع) إلا أنه شمل في تصميم هذه الآلة العديد من أساسيات الكمبيوتر الحالية. وقد احتوت آلة التحليل على مخزن لحفظ عنصر بيانات وآلة للعمل عليه. وقد كان باباج منبهراً جداً بعمل جاكورد بالبطاقات المثقبة. وقد كانت آلة التحليل قادرة في واقع الأمر على تخزين تعليمات الحسابات في بطاقات مثقبة والتي يمكن أن تغذي في الآلة مع بطاقات مثقبة تحتوي على بيانات لتعمل على هذه البيانات وتنتج النتائج المرجوة.

اجراء الحسابات كبيرة الحجم عملياً

تطلب تطوير آلات اجراء الحسابات وعمل الجداول شيئين:

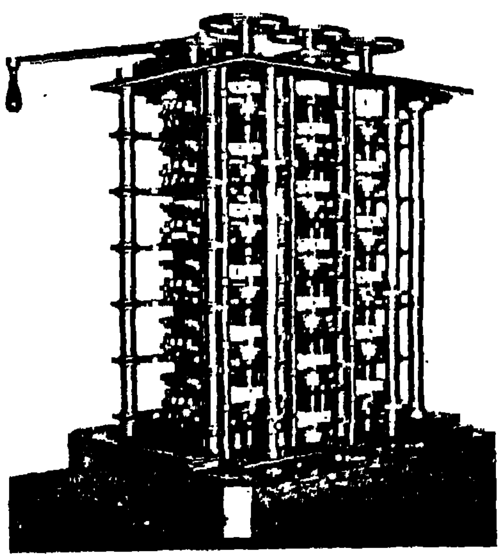
١ ـ مخزن بيانات ومتطلبات تشغيل كبيرة الحجم.

٢ ـ مستوى أساسي معين من مهارات التصنيع وآلات ميكانيكية وكهربائية على مستوى معين من الدقة.

وفي أواخر القرن التاسع عشر الميلادي وصلت مهارات التصنيع مستوى متطور كاف كما أن الحاجة الى كميات هائلة من البيانات حدثت كذلك: تعداد سكان الولايات المتحدة الأمريكية.

لقد أخذ اجراء تعداد سكان الولايات المتحدة الأمريكية عام 1880 م سبعة سنوات لاتمامه يدوياً. وقد قدر أن استخدام نفس الوسائل اليدوية في عمل تعداد عام 1890 م لن ينتهي إلا بعد البدء في جمع بيانات تعداد عام 1900 م. وقد قدم أحد مهندسي الحكومة والمسمى هيرمان هوليريث Herman Hollerith حلاً لمشاكل تشغيل بيانات تعداد السكان. واعتماداً على مفهوم جاكورد للبطاقة المثقبة





شكل 1.7
باباج وآلة العزف التي شيدها
(بإذن من أرشيف IBM)

(كما فعل باباج) فقد رتب هوليريث لاعداد بيـانات التعـداد بتخزينهـا في بطاقـات مثقبة. وقد بنى آلات لتثقيب الثقوب في البطاقات وآلات لترتيب البطاقات شكـل 1.8. والفرش السلكية التي تمس الثقوب توصل التيار الكهـربائي عنـدما تمـر على الثقوب الموجودة في البطاقات وتغير من قيم عدادات ميكانيكية كهربائية. وقـد تم عمل تعداد عام 1890 م بعد شهر من الانتهاء من جمع البيانات وذلك بـاستخدام آلات هوليريث.

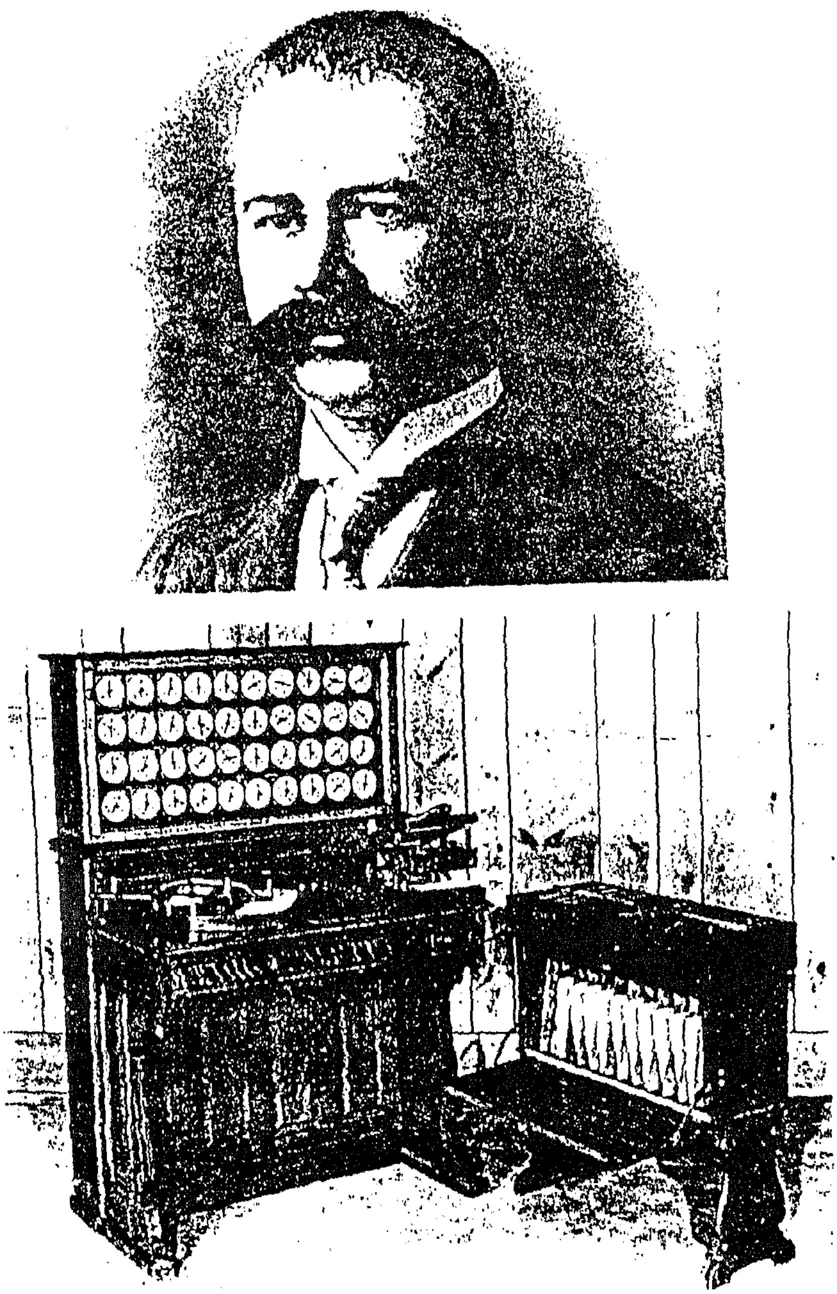
وقد أخذت المجمـوعة الكـاملة للجداول والتي شملت بيـانات عن أسئلة لم يكن من العملي سؤلها قبل ذلك سنتين.

وقـد كـوّن هـولـيريث عـام 1896 م شركـة لآلات الجــدولـة Tabulating Machine Company لانتـاج وتسويق آلاتـه تجاريـاً. وهذه الشركـة بـدمجهـا مـع شركات أخرى كونت ما يسمى حالياً بشركة آلات الأعمال الدولية International Business Machines Corporation (IBM).

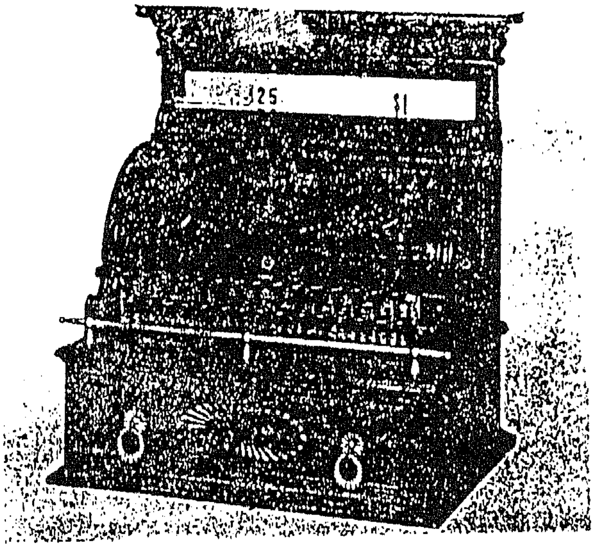
وقد استمر مكتب عمل تعداد السكـان الأمريكي أثنـاء استخدامـه معدات هوليريث في إجراء التجارب الخاصة به. وقد طور أحد مهنـدسيه ويـدعى جيمس بورز James Powers آلات تغذي البـطاقات في المعـدات تلقائيـاً وكذلـك تطبـع النتائج آلياً. وقد كون عام 1911 م شركة آلات جدولـه بورز Powers Tabulating Machine Company والتي كونت أساس وحده UNIVAC في شركة سبيري Sperry Corporation.

وأثناء هذه الفترة أيضاً حدث تقدم أساسي في تتبع البيانات ميكـانيكياً فقـد تم تسويق مسجل النقد بشكل كبير جداً عن طريق شركة مسجلات النقد القومية National Cush Register Company. وقـد كان مسجـل النقد قـد اخترع عـام 1879 م واكتسب في هذه الفترة قبـولاً كبيراً في أعـمال البيع بـالتجزئـة بكل المعـاني شكل 1.9.

ومع تزايـد درجة تعقيـد المجتمع ازداد كـذلك حجم بيـاناتـه وتعقيد طـرق تشغيلهـا. وقد قـدم المخترعـون تقدمـات عديـدة كـاستجـابـة لـذلـك. وخـلال العشرينيات والثلاثينيات من القرن الحالي قام فانيفار بـوش Vannevar Bush من معهـد ماسـاشوستس للتقنيـة Massachusetts Institute of Technology (MIT)



شكل 1.8
هوليريث وبعض من معداته للبطاقات المثقبة (بإذن من أرشيف IBM)



شكل 1.9
مسجل نقد قديم من NCR (بإذن من NCR Corp)

وآخرون من معاهد مماثلة مثـل مدرسـة مور للهنـدسة بجـامعة بنسلفـانيا Moore School of Engineering of the University of Pennsylvania بتـطويـر أجهـزة كمبيـوتر تمـاثلية analog تعتمـد عـلى آلات قيـاس مستمـرة عـلى عكس العنـاصر المتقـطعة مثـل الثقوب المـوجودة في البـطاقـات وكـان ذلـك لحسـابـات القـذائف العسكرية في البداية .

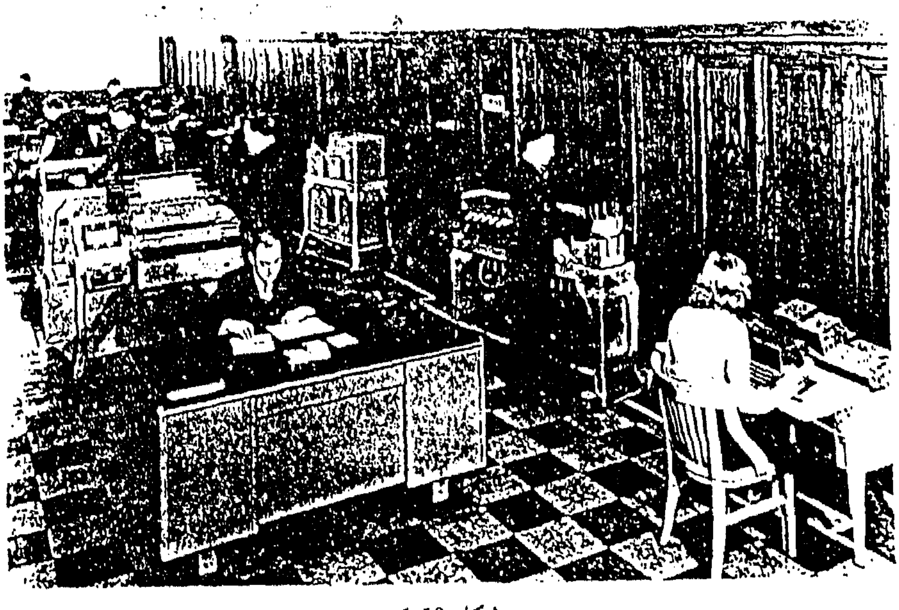
وفي نفس الفترة ومع الاستمرار في بداية الأربعينيات الميلادية من هذا القرن كان تشغيل البيـانات تجـارياً يؤدي عـلى العديـد من الآلات الميكانيكيـة كهربـائية المعتمـدة على البـطاقات المثقبـة . وقد شملت الحـاسبات والشـاقبات وآلات الفـرز وآلات الترتيب والطابعات، شكل 1.10 . وكانت تخزن البيـانات في بـطاقات مثقبـة بينما كانت تنفذ تعليمات التشغيل كتجمعات من الأسلاك توصل في لوحـة مصممة خصيصـاً لـذلـك والتي يتم ادخـالهـا بـدورهـا في فتحـات في الآلات الميكـانيكيـة كهربائية .

ومـع الوصـول للحرب العـالمية الثـانية أسرعت خـطوات التطور فقـد قـام جيورج ستيبتز George R. Stibitz في معـامل Bell للهـاتف وهوارد ايكـين Ho-

ward H. Aiken بجامعة هارفارد Harvard بانتاج أجهزة كمبيوتر رقمية ميكانيكية كهربائية معقدة. وقد تم تصنيع جهاز ايكين Mark I عام 1944 م وكان به 72 عداداً عددياً أو مواقع تخزين وكان يتطلب 5 ثوان لأداء عملية الضرب.

كما إن أول جهاز كمبيوتر اليكتروني كبير كامل ظهر كذلك طبقاً لاحتياجات القذائف العسكرية أثناء الحرب. وقد بدأ العمل فيه عام 1943 م في مدرسة مور Moore School وعمل فيه جون موشلي John W. Mauchly وبرسبر ايكرت Presper Eckert وسمى حاسب ومكمل عددي اليكتروني Electronic Numerical Integrator and Calculator (ENIAC) وتم تصنيعه عام 1946 م شكل 1.11. وقد كان يحتوى على 18,000 صمام مفرغ وكان بإمكانه أداء 5000 عملية جمع في الثانية الواحدة. وكان أسرع بعدد 1000 مرة من Mark I.

ومع حدوث التقدم في أجهزة الكمبيوتر الألكترونية الكاملة حدث كذلك تطورات أخرى تبعتها. ففي عام 1945 م قام جون فون نيومان John Von Newmann بعمله على أفكاره وأفكار غيره باقتراح كمبيوتر يخزن فيه البرامج. وقد



شكل 1.10

مثقبات بطاقات ميكانيكية كهربائية مستخدمة كمعدات تشغيل بيانات من حوالي 1940 م

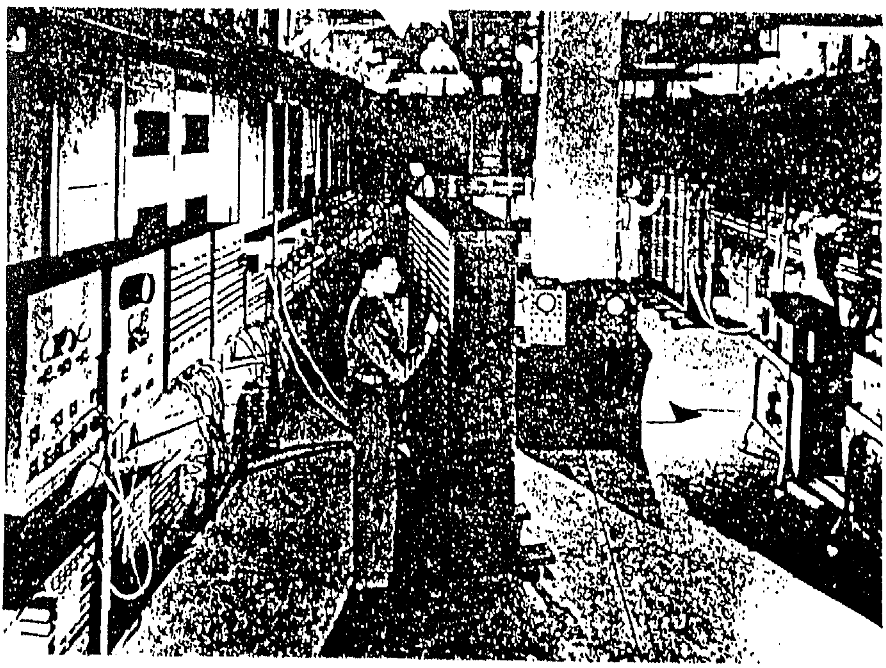
(بإذن من أرشيف IBM)

سميت أول آلـه من هذا النـوع بكمبيوتـر آلي للتخزين الألكـتروني. Electronic Delay Storag Automatic Computer (EDSA).

وقـد تم إعداده عـام 1949 م في جامعـة كمبردج في انجلترا وأتمـه موريس ويلكس Maurice Wilkes. وفي عام 1952 م كان جهاز Whirlwind I من MIT والذي سبق تقديمه في نهاية الأربعينيات أول جهاز يحتوي على ذاكرة قلوب.

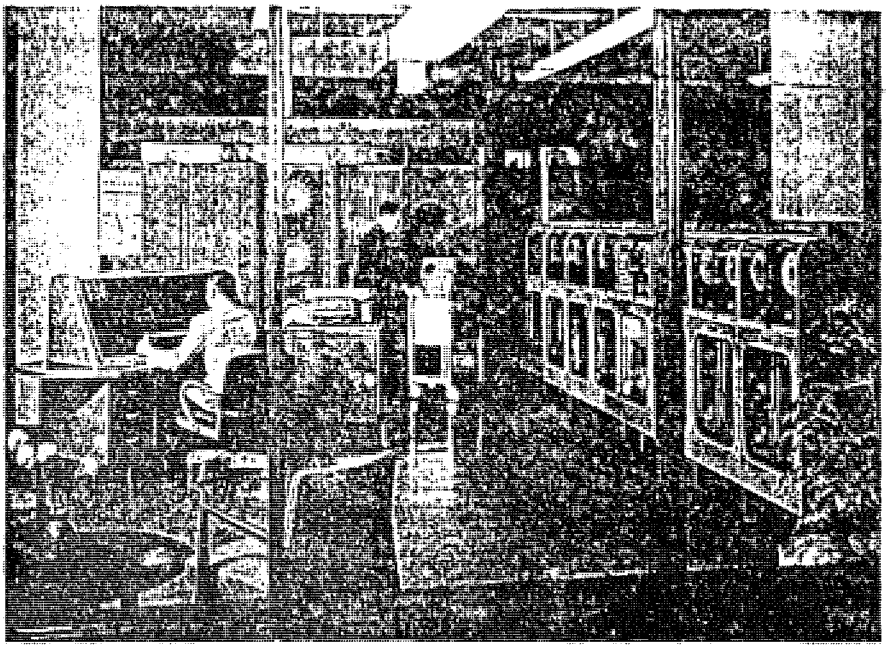
أجهزة الكمبيوتر التجارية

بعد الحرب العالمية الثانية فإن التطورات والـزيادات في احتيـاجات الأعـمال التي وجدت بسبب التوقعات الاجتماعية تسببت في ازديادات أكثر في أنواع وأحجام البيانات المـراد تشغيلها. وقـد أوجد ذلـك ضغوطـاً محددة عـلى امكانيـات تشغيل البيانات وقد شكل إيكـارت Eckert وموشـلي Mauchly شركة كمبيـوتر إيكـارت ومـــوشـلي Eckert - Mauchly Computer Corporation عــام 1947 م. وقــد أنتجت الشركة عام 1951 جهاز كمبيوتر المسمى الكمبيوتر الشامـل الآلي Universal Automatic Computer (UNIVACI) شكــل 1.12 وذلـــك لمكتب تـعــداد



شكل 1.11

جهاز ENIAC (بإذن من Sperry Corp)

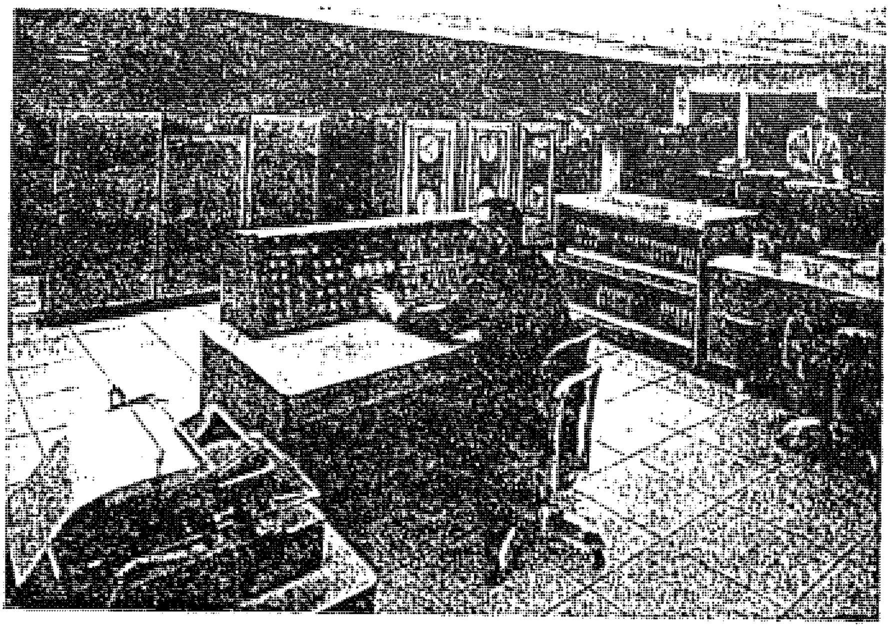


شكل 1.12
جهاز UNIVAC 1 (بإذن من Sperry Corp)



شكل 1.13
جهاز IBM 705 (بإذن من أرشيف IBM)

السكان الأمريكي. وقد أدمجت هذه الشركة في شركة Remington Rand والتي سميت فيما بعد بشركة سبيري Sperry Corporation واستمرت في انتاج أجهزة الكمبيوتر تحت اسم UNIVAC. وفي عام 1953 قامت شركة IBM التي كان لها وجود قوي بالنسبة لمعدات عمل الحسابات ميكانيكياً وكهربائياً ومعدات معتمدة على الصمامات المفرغة بانتاج أول جهاز كمبيوتر اليكتروني لها. وقد كان الكمبيوتر IBM 701 آلة موجهة للأغراض العلمية ولها ذاكرة من الصمامات المفرغة وقد تبعها جهاز 702 عام 1955 م والذي كان له ذاكرة من القلوب شكل 1.13. هذه الآلات وغيرها من الآلات الشبيهة عرفت بأنها الجيل الأول للكمبيوتر.



شكل 1.14
جهاز كمبيوتر من الجيل الثاني **NCR 304** (بإذن من **NCR Corp.**)

وقد شاهدت نهاية الخمسينيات وبداية الستينيات الميلادية ميلاد أجهزة الكمبيوتر المعتمدة على الترانزستور على عكس الصمامات المفرغة شكل 1.14. وهذه الآلات أصبحت الجيل الثاني للكمبيوتر وقد كانت آلات يمكن الاعتماد عليها أكثر وكانت أصغر في الحجم وأسرع في أداء الحسابات عن آلات الجيل الأول. وكانت هذه التقنية الجديدة بالإضافة إلى إدخالها في أجهزة الكمبيوتر الكبيرة بدمجها مع تحسينات في تقنية ذاكرات القلوب سبباً في إنتاج أول أجهزة



شكل 1.15
جهاز كمبيوتر من الجيل الثالث جهاز UNIVAC 1108 (بإذن من Sperry Corp)

كمبيوتر صغيرة الحجم والتي أصبحت ناجحة جداً وانتشرت بمعدل سريع.

وقد كانت هذه الآلات أول أجهزة كمبيوتر للعديد من آلاف الشركات وقد شاهد منتصف وأواخر الستينيات الميلادية ظهور الجيل الثالث للكمبيوتر. وقد كان الجيل الثالث للكمبيوتر المبني على الدوائر المتكاملة خطوة رئيسية أخرى للإمام شكل 1.15.

أوساط التخزين الحديثة للبيانات

على التوازي مع نمو معدات تشغيل البيانات كان هناك تطور في وسط جديد لتخزين البيانات. والصورة المبكرة للتخزين الحديث للبيانات كانت الثقوب الموجودة في شرائط ورقيه مناسبة. وقد ظهرت الشرائط الورقية المثقبة في السبعينيات والثمانينيات من القرن الماضي وذلك بالاتصال مع المعدات من نوع التلغراف. وبالطبع سبق أن ذكرنا أن هوليريث في التسعينيات من القرن الماضي وبورز في بداية القرن الحالي استخدما البطاقات المثقبة كوسط تخزين. وفي واقع الأمر فإن البطاقات المثقبة كانت وسط التخزين المستخدم الوحيد في آلات المحاسبة الآلية متزايدة التعقيد في العشرينيات والثلاثينيات والأربعينيات الميلادية من القرن الحالي وقد كان استخدامها كبيراً في أجهزة الكمبيوتر المبكرة التي وجدت في الخمسينيات والستينيات الميلادية كما أنها استخدمت أيضاً في

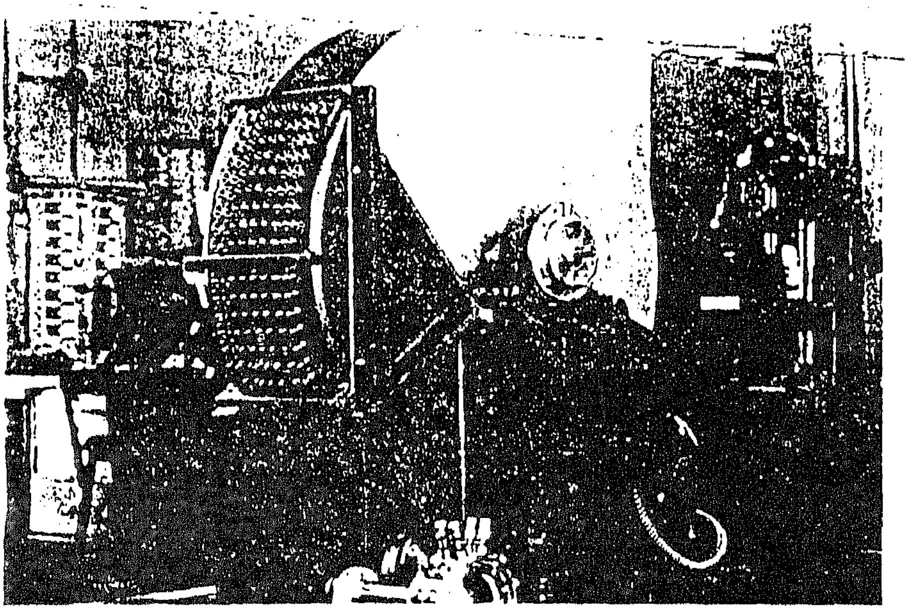
السبعينيات الميلادية في مؤسسات تشغيل البيانات الصغيرة ولكن بكمية أقل في الواقع.

في عام 1936 م قام ايستمان كوداك Eastman Kodak وبوش Bush باستخدام المختار السريع Rapid selector الخاص بتسجيل بيانات بالرسم التصويري. بينما يمكن إعادة قراءة البيانات بتقنية الخلايا الضوئية الكهربائية إلا أن هذا الوسط غير قابل لإزالة البيانات من عليه. وبينما يمكن أن تكون سمة عدم القابلية للإزالة في وسط تخزين البيانات من المميزات بالنسبة لبعض تطبيقات تشغيل البيانات إلا أن ذلك يكون محدوداً جداً لمعظمها وعلى هذا فإن استخدامات تخزين الرسم التصويري لم تكن سوى استخدامات محدودة دائماً.

وقد شاهد منتصف الثلاثينيات الميلادية بداية عصر أوساط التخزين المغناطيسي وذلك بإجراء معامل بل للهاتف تجارب على شريط مغناطيسي لتخزين الصوت. وفي عام 1937 م كان هناك عمل مبكر في المانيا على الأكاسيد لاستخدامها في التخزين المغناطيسي. وفي عام 1937 م وعام 1938 م عبر بوش Bush عن بعض الأفكار المبكرة الخاصة بالتخزين المغناطيسي على أشياء تشبه الألواح. وفي عام 1939 ربطت MIT شرائط مغناطيسية تخلف بطاقات المحاسبة وذلك حول اسطوانة لقراءتها. ومع الحرب العالمية الثانية قلت أو توقفت هذه المجهودات.

وقت كتب بيري كراوفورد Perry Crawford في MIT عام 1942 بحثاً يصف فكرة تخزين الاسطوانة المغنطة. وقد أجرت شركة الأبحاث الهندسية Engineering Research Associates (ERA) تجارباً مع هذه الآلات وطورتها في 1943 م وفي بداية الخمسينيات الميلادية شكل 1.16.

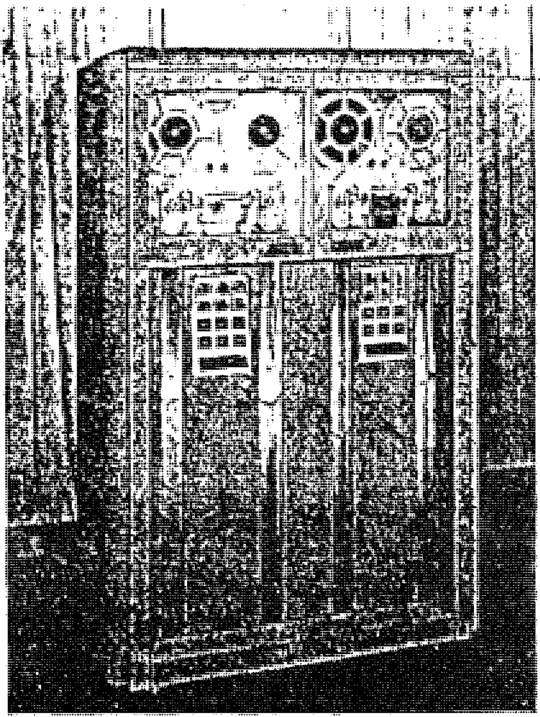
وفي أواخر الأربعينيات الميلادية كان هناك عمل مبكر عن استخدام الشريط المغناطيسي في تسجيل البيانات وأفكار كثيرة عن التخزين المغناطيسي للأسطح الأفقية. وقد طور إيكرت وموشلي بصفة خاصة عام 1947 م وحده شريط مغناطيسي. واستمرت شركة IBM في تطوير الشريط المغناطيسي عام 1948 م وعام 1949 م كما أن أفكار الأسطح الأفقية والأسطح التي لها شكل كعكي donut الخاصة بتخزين البيانات للاتصال العشوائي كبير الحجم تم التعبير عنها كذلك عام



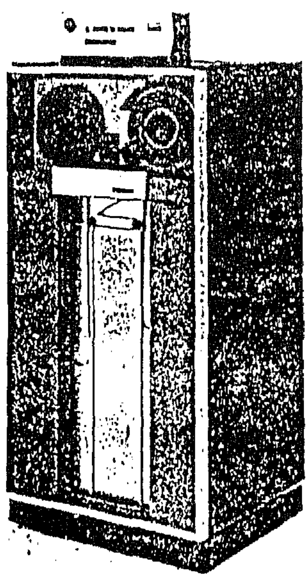
شكل 1.16
وحده اسطوانة مغناطيسية قديمة للتخزين، عام 1948 م
(بإذن من Control Data Corp)

1947 م في Aberdeen Proving Grounds وفي المكتب القومي للقياسات National Bureau of Standards على التوالي وفي عامي 1948 م و 1949 م أجرت Potter Instrument تجارب على شرائح من الشريط المغناطيسي مشدودة على هيكل أفقي.

وفي عام 1950 كان هناك آخرون منهم RCA و Raytheon طوروا مفهوم الشريط المغناطيسي كذلك. وفي بداية الخمسينيات طور ايكين كذلك وحدات شرائط بأعمدة فارغة. وفي عام 1952 قدمت كل من UNIVAC و Raytheon وحدات شرائط مغناطيسية بصورة تجارية. وقدمت IBM عام 1953 وحدات شرائط شكل 1.17. ومنذ منتصف الخمسينيات وحتى منتصف الستينيات أصبحت الشرائط مسيطرة على أوساط تخزين البيانات للكمبيوتر. وقد تحسنت تقنية الشريط المغناطيسي بصورة مستمرة بعد ذلك وما زالت تستخدم في نطاق واسع الآن وبصفة خاصة في بيانات الأرشيف. شكل 1.18.

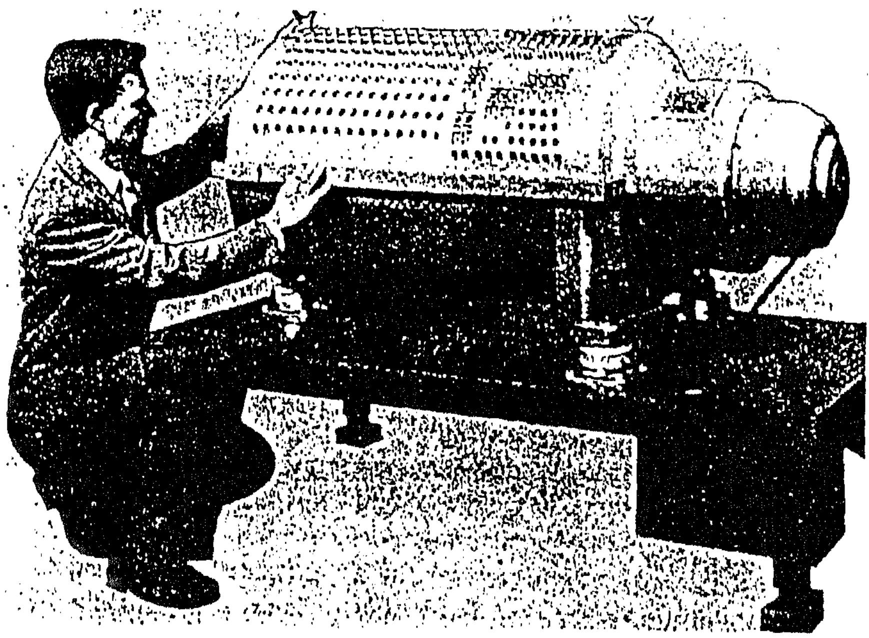


شكل 1.17
مشغل شرائط قديم IBM
(بإذن من أرشيف IBM)



شكل 1.18
مشغل شرائط أحدث
(بإذن من أرشيف IBM)

وفي عام 1952 م وعام 1953 م قدمت شركة ERA وتبعتها شركة UNIVAC أول اسطوانات مغناطيسية شكل 1.19. وقدمت شركة IBM في عامي 1953 م و 1954 م اسطوانات مع بعض أجهزة الكمبيوتر المبكرة الخاصة بها. وقد لحقتها في هذا المضمار شركة Control Data عام 1955. كما أن File Computers من UNIVAC عام 1955 احتوت على أول تخزين اسطوانات له سعة كبيرة.



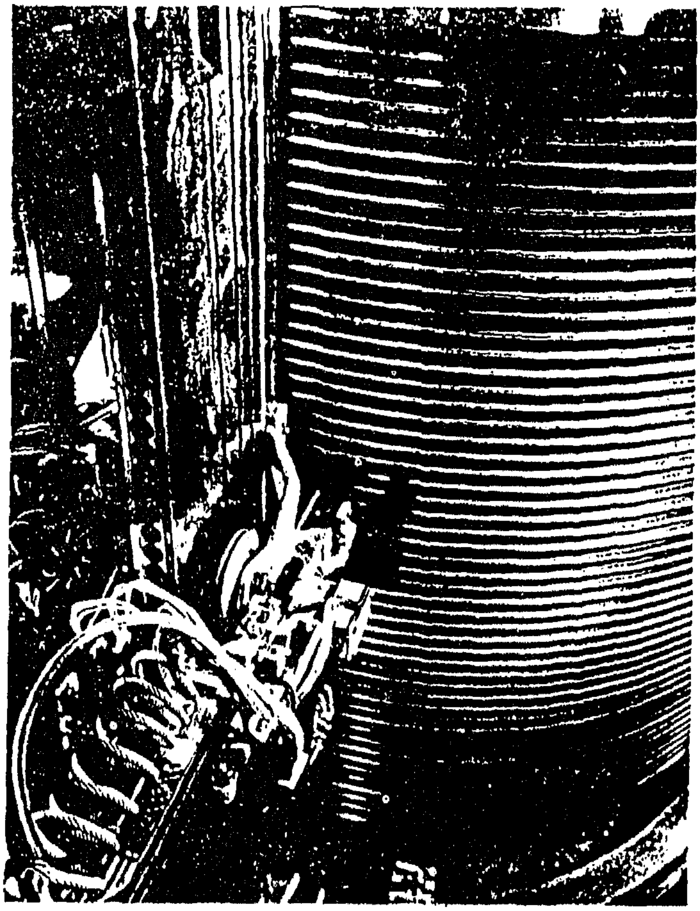
شكل 1.19

وحده اسطوانة ممغنطة مطورة ERA (بإذن من Control Data Corp)

وقد طورت شركة IBM عامي 1956 م و 1957 م بطاقات مغناطيسية كوحدات ذاكرة للآلات الكاتبة.

وبدأت شركة IBM عام 1953 م في العمل على وحدة تخزين أقراص ثابتة لآلة المحاسبة ذات ذاكرة الاتصال العشوائي. Random Access Memory Accounting Machine (305 RAMAC).

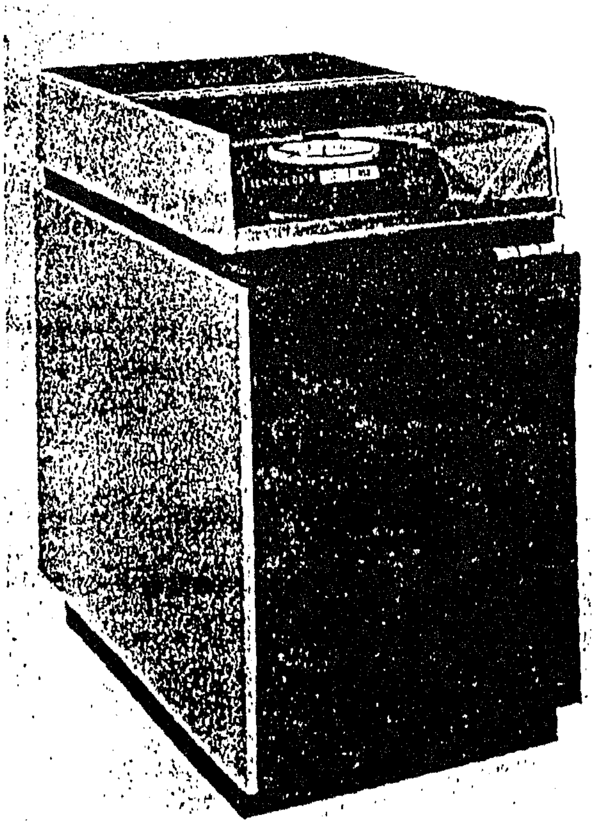
وفي عام 1954 م كان هناك العديد من الصيغ التي أصبحت متاحة تجارياً عام 1956 م شكل 1.20. وفي عام 1957 م أعدت أول شحنة من وحدات تشغيل



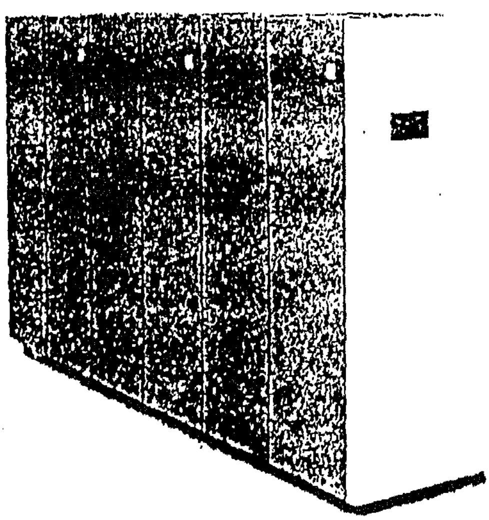
شكل 1.20 مشغل أقراص IBM RAMAC (بإذن من أرشيف IBM)

الأقـراص ذات الرؤوس المتحـركة والتي قـدمت في الأسواق عـام 1963 م، شكل 1.21. وخلال منتصف الستينيات بـدأ يحدث تحـول كبير من الشرائط المغنـاطيسية كوسط تخزين ثانوي إلى الأقراص المغناطيسية.

في البداية كانت وحدات تخزين الأقراص مقيـدة ناحيـة فلسفة المجمـوعات التي يمكن نقلهـا مع نسبـة من دستة مجمـوعات أو أكـثر توضـع على مشغـل واحد وتـزال من عليه. ومـع ازدياد كثـافة تسجيـل البيـانـات قلت نسبـة المجمـوعـات للمشغل الى النقطة التي سوقت فيها شركة IBM عام 1976 م مشغـل أقراص ومعـه مجمـوعة غـير قابلة للنقـل. وقد قـاد التحسـين المستمـر في تقنيـه القـرص إلى أنها أصبحت وسط التخزين النمطي للبيانات حالياً شكل 1.22.



شكل 1.21
أحد مشغلات الأقراص في منتصف الستينيات الميلادية
(بإذن من Control Data Corp)



شكل 1.22
وحده مشغل أقراص حديث ومحكمة
(بإذن من .IBM Corp)

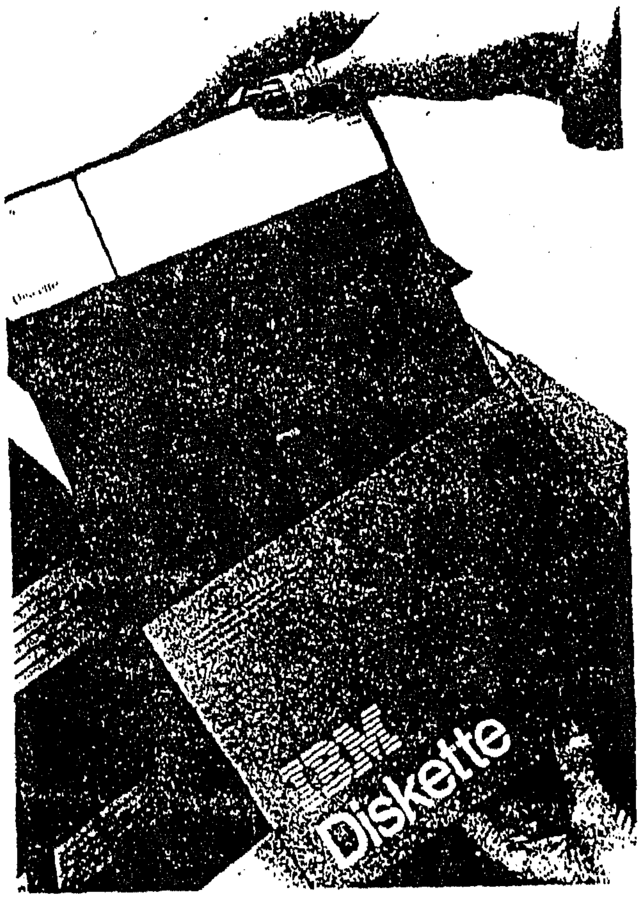
ومن ضمن أحدث أوساط تخزين البيانات توجد وحدات التخزين الغزير Mass Storage والقريصات. وقد قدمت شركة IBM عام 1974 م نظام تخزين غزير يحتوي على شكل يشبه خلية النحل به اسطوانات صغيرة ملفوف حولها شرائط مغناطيسية شكل 1.23 وطبقاً للطلب يمكن للوحدة أن تختار أسطوانة وتلف شريطها حول وحدة القراءة وتنقل البيانات. وبالرغم من بطء هذه التقنية بالمقارنة بالأقراص إلا أنها قادرة على تخزين كميات هائلة من البيانات. وفي الطرف الآخر فإن الأقراص الصغيرة (القريصات) صغيرة الحجم ومرنة وغير مكلفة ومغطاة بمادة بلاستيكية وقد قدمت هذه القريصات في صور مختلفة عام 1971 م، وعام 1973 م



شكل 1.23

اسطوانات شرائطا نظام IBM للتخزين الغزير (بإذن من .IBM Corp)

شكـل 1.24. ومشغلات القـريصات هي وحـدات التخزين الثـانـوي الأسـاسيـة لأجهزة الميكروكمبيوتر في وقتنا الحالي.



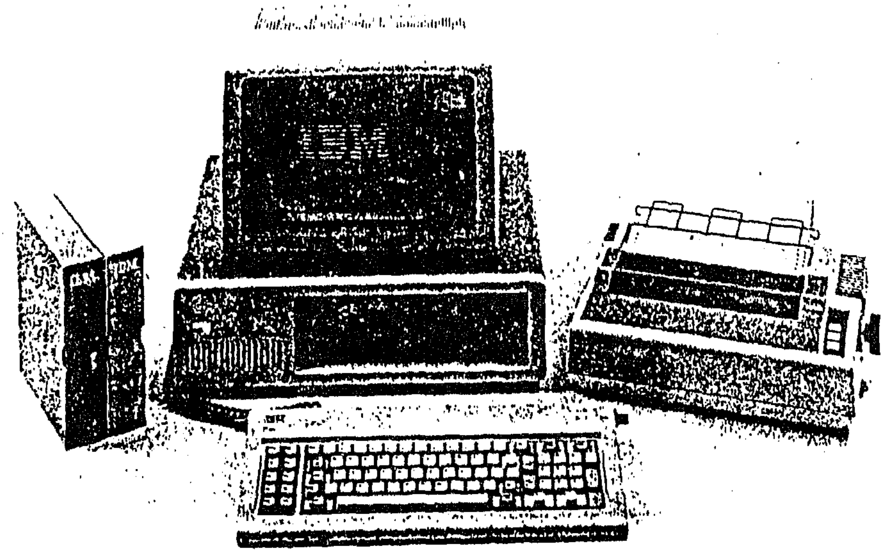
شكل 1.24
قرص مرن أو قريص (بإذن من .IBM Corp)

بيئة تشغيل البيانات في الوقت الحالي

ربما تصبح الحدود بين أجيال الكمبيوتـر غير مميـزة. ونحن نتكلم عن الجيل الرابع للكمبيوتر حالياً على أنه مبني على أساس دوائر متكاملة عـلى مستوى واسـع جـداً Very large scale integrated circuits (VLSI). هذه التقنيـة والتحسينات

الأخرى غير المعدودة استمرت مع الاتجاه المتزامن لقلة الأسعار وزيادة الأداء.

وتتراوح أجهزة الكمبيوتر حالياً من أجهزة الكمبيوتر المنزلية أو الميكروكمبيتور التي يمكن أن توضع فوق المنضدة شكل 1.25 إلى آلات لها قدرات



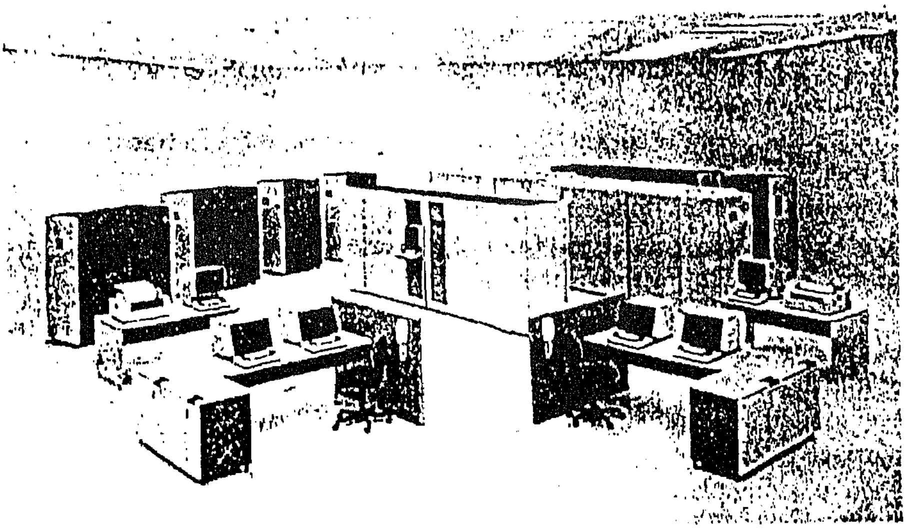
شكل 1.25

جهاز ميكرو كمبيوتر: جهاز IBM PC (بإذن من .IBM Corp)

هائلة شكل 1.26. وقد أصبحت تسهيلات التخزين أسرع ويمكن الاعتماد عليها أكثر وقادرة على حفظ البيانات في صورة تتزايد كثافتها. كما انتشرت شبكات الكمبيوتر المرتبطة مع بعضها بواسطة أوساط اتصالات مختلفة.

كما أن نظم البرامج وهي الأنواع المختلفة من التعليمات التي تجعل الآلة تعمل، استمر التقدم فيها كذلك. وتسمح نظم التشغيل مرتفعة التعقيد للعديد من الناس باستخدام الآلات في نفس الوقت. وتسمح نظم إدارة قواعد البيانات لتخزين غير متكرر للبيانات ووصول لبيانات معقدة وتسمح نظم التطوير الخاص بأعداد برامج تطبيقات خاصة بسرعة مع تدمير عملية البرمجة التقليدية البطيئة.

وإيجازاً أصبح الكمبيوتر جزءاً من كل عملية أعمال فاضلة ومن المؤكد أن هذا الاتجاه سوف يستمر.



شكل 1.26
جهاز كمبيوتر حديث وكبير: جهاز IBM 3084 (بإذن من .IBM Corp)

عصر المعلومات The Information Age

ماذا أعادنا إلى نقطة بداية كل هذا النشاط: البيانات. لقد قضى الناس آلاف من السنين لزرع الاهتمام بأبسط أنواع البيانات وقد مضى عدة مئات من السنوات في تطوير طرق ووحدات عمل الحسابات الأساسية وذلك الى نقطة أصبح عندها الاهتمام باستخدامات أكثر تعقيداً غير عملي. وقد مضى حوالى 30 سنة فقط ليصل تشغيل البيانات الى نقطة يتحدث عندها الناس عن انحسار في عصر الصناعة وميلاد عصر المعلومات.

أحد الأوجه الحاسمة لعصر المعلومات وهو أحدها الذي يجب أن نبدأ به هو إدارة الأحجام الهائلة للبيانات المشملوة. امش في إحدى منشآت تشغيل البيانات لشركة كبيرة في وقتنا الحالي وسوف ترى أرففاً موجود عليها بكرات للقراءة فيها والكتابة عليها. وبالطبع فإن التطورات الجديدة في نظم المكونات تستمر في إنتاج وحدات يمكنها أن تحتوي على بيانات أكثر في نفس المكان. ولكن بطريقة تضيف للمشكلة بدلاً من حلها وذلك لأنها تشجع الناس على اكتساب البيانات وتخزين الكثير منها ذلك بدون وجود وسيلة كافية لإدارتها بطريقة مناسبة.

كميـة البيانـات الموجـودة في المؤسسات الكبـيرة وحتى غير الكبـيرة في وقتنـا الحالي هي كمية لا يمكن تصديقها. فنحن نتكلم عن بلايين أو ترليونات وأكثر من قطع فردية من البيانات. والأكثر اعتياداً الطريقـة التي تدار بهـا أو بدقـة أكثر سـوء الإدارة فهي رهيبة إذا كانت مفهومة بعض الشيء وهذا ليس كما لـو كنا نتكلم عن إدارة العاملين بالشركـة. حتى الشركات الكبـيرة لديهـا عدة مئـات من الآلاف منهم ولا يتكرر تغييرهم جميعاً. أو النقود التي تملكها الشركة وبـالتأكيـد هناك كم هـائل منهـا إلا أنها جميعهـا نفس الشيء بمعنى أنها دولارات تـدفـع كـرواتب وهي مثـل الدولارات التي تدفع للموردين ثمناً للمواد الخام.

وفي البيانات يكون لدينـا سلعة لهـا حجم هائـل وكل جـزء منها مختلف عن الآخر. ولها خاصية وهي أن معظمها في حالة تغـير في أي وقت واحد. وفي وقتنـا الحالي يبدو أن كل فرد في الشركة يريد اتصال (عـادة اتصال لحـظي) بها لسبب أو لآخـر. وأكثر من ذلـك مع حـدوث هذا التـطور الكثير في تشغيـل البيانـات فمن الصعب تتبع البيانات الموجودة بالفعل. لهذا السبب ولأسباب أخرى تميل البيانات الى أن تخزن العديد (أو عدة مرات) من المرات بالنسبة للتطبيقات المختلفة.

أليس هناك أمل؟ هـل يجب علينا أن نـدخل في مجـال شائـك لكم لا نهائي من البيانات غير المدارة للأبد؟

لحسن الحظ هنـاك فئة من الـطرق ـ في الـواقـع أكـثر من بيئـة ـ لتخـزين البيانات والاتصال بها وإدارتها والتي يمكن أن تشكل قـاعدة لـوجود عـالم يمكن أن يصبح موجهاً أكثر للبيانات مع مرور الوقت. وفئة الطرق أو البيئة أصبـح معروفـاً بأنه قاعدة البيانات.

هذا الكتاب This Book

لقـد وجدت قـاعـدة البيـانـات كمـوضـوع وكفـرع علمي مميـز في منتصف الستينيـات إلا أنه كـان محدوداً لبعض الـوقت في الاهتمام بجـزء صغـير نسبيـاً من مجموع تشغيل البيانات. وقـد نمى خلال السبيعينيـات ليصبح بؤرة رئيسيـة لنشاط تشغيل البيانات في العديد من المنظمات. ومع المزيد والمزيد من العاملين في تشغيل البيانات وتعرية البيانات فيتوقع لهم أن يفهموها ويستخدموها. وقد أصبح واضحاً

أنه هناك افتقار هائـل في فهم المفاهيم الأسـاسية للمـوضوع. ودخـول الثمانينيـات جعل الموقف أكثر حدة.

إن مفهوم قاعدة البيانات مفهوم قوي جداً. فهو يشكل حـلاً داخلياً لبعض المشاكل المثيرة. ونتيجة لذلك فإنه يميل لسوء الحظ أن يكون متعدد الأوجـه ومعقد ويبدو صعباً عندما يحـاول الشخص أن يبلغه مـرة واحدة. إلا أنـه يمكن الوصـول إليه وفهمه إذا مـا استمر الشخص فيـه بعنايـة وحرص وتقـدم في الموضـوع خطوة خطوة.

وهذا الكتاب مقدمة مباشرة لقواعد البيانـات. ويأخـذ منهج الخـطوة خطوة الذي لا يترك فراغات في توضيحاتـه. ويضيف كل فصـل المزيـد من التقدم لفهم أساسيات قـواعـد البيـانـات. ومن المفهـوم أنـه لا يمكن لأي شخص في تشغيـل المعلومات أن يستمر في عمله بدون ذلك.

مراجع References

Crawford, P., conversation on February 3, 1984.

Gleiser, M., «Men and Machines Before Babbage», DATAMATION, vol. 24, no. 10, October, 1978, pp. 125-130.

Goldstine, H. H., The Computer from Pascal to von Neumann, Princeton University Press, Princeton, NJ, 1972.

Gordon, M. J., and Shillinglaw, G., Accounting: A Management Approach, 4th ed., Richard D. Irein, Homewood, IL, 1969.

Hendriksen, E. S., Accounting Theory, revised ed., Richard D. Irwin, Homwood, IL, 1970.

Littleton, A. C., and Yamey, B. S., Studies in the History of Accounting, Richard D. Irwin, Homewood, IL, 1956.

Rosen, S., «Electronic Computers: A Historical Survey», Computing Surveys, vol. 1, no. 1, March, 1969, pp. 7-36.

Schmandt-Besserat, D., «The Earliest Precursor of Writing», Scientific American, vol, 238, no. 6, June, 1978, pp. 50-59.

Shurkin, J., Engines of the Mind, W. W. Norton, New York, 1984.

Stevens, L.D., «The Evolution of Magnetic Storage», IBM Journal of Research and Development, vol. 25, no. 5, September, 1981, pp. 663-675.

Tropp, H., «The Effervescent Years: A Retrospective», IEEE Spectrim, vol. II, no. 2, February 1974, pp. 70-79.

أسئلة وتمارين Questions and Exercises

١ ـ ارسم خـطاً للوقت خاص بـآخر مـائتي سنة يقـارن تطويـر وحدات تشغيـل البيانات مع تطوير أوساط تخزين البيانات.

٢ ـ هـل تعتقد أن التـطور المقارن لـوحدات تشغيـل البيانـات، وأوسـاط تخـزين البيانات كان غير زوجي؟ هل سرعة نمو أحدهما أثرت في نمو الآخر؟.

الفصل الثاني
الملفات البسيطة وأوساط التخزين
Simple Files and storage Media

ما البيانات؟ What is Data?

السجلات والكينونات

دعنا نبدأ بالتفكير في جزء من البيانات. يمثل الجزء الفردي من البيانات حقيقة فردية عن شيء معين نهتم به. فيمكن أن تكون في أحد الشركات الحقيقة بأن رقم العامل John Smith هو 31719 أو يمكن أن تكون الحقيقية بأن أكبر مورد للمسامير للشركة يوجد في شيكاغو أو يمكن أن تكون الحقيقة بأن رقم هاتف أحد عملاء الشركة وهي شركة Acme Company هو 555-8003. وجزء البيانات Piece of Data هو حقيقة fact فردية عن شيء معين نهتم به فيما يحيط بنا.

وعادة ما يكون لدينا العديد من الحقائق التي تصف شيء معين يهمنا ويقع في بيئتنا. مثال ذلك دعنا نعتبر الحقائق التي يمكن أن نكون مهتمين بها والخاصة بالعامل John Smith ونريد أن نتذكر أن رقم هذا العالم هو 31719 وإن معدل راتبه في الساعة هو 5.50 $ وإن بلدته هي ديترويت ووليـاتها ميتشجان وما إلى ذلك. نحتاج إلى أن نعرف هذه الأشياء لكي نقوم بتشغيل شيك الراتب لهذا العامل كل أسبوع وللاتصال به في حالة إذا تأخر عن العمل أو الاتصال بأسرته إذا احتاج العمل أن يتأخر به ولإرساله إلى الشركة التي تعد بطاقات التهنئة بالأعياد وما إلى ذلك. ويبدو بكل تأكيد أنه من المعقول أن تجمع كل الحقائق عن John Smith التي نحتاجها لهذه الأغراض مع بعضها وتحفظ جميعها. يبين شكل 2.1 كل هذه الحقائق عن John Smith والتي تهم الراتب والتطبيقات المتعلقة به.

Employee Number	Employee Last Name	Employee First Name	Salary Rate	Home city	Home state	Home Telephone
31719	Smith	John	5.50	Detroit	MI	313-555-5186

شكل 2.1
حقائق عن راتب John Smith

حيث أننا سنعد شيك كل أسبوع لكل عامل في الشركة وليس لهذا العامل John Smith فقط فنحن في حاجة حتماً إلى تجميع لحقائق تشبه حقائق الشكل 2.1 لكل من العاملين في الشركة. شكل 2.2 يبين جزء من هذا التجميع.

Employee Number	Employee Last Name	Employee First Name	Salary Rate	Home city	Home state	Home Telephone
06337	Jones	Fred	4.30	New York	NY	212-555-9207
09155	Smith	Susan	5.90	Atlanta	GA	404-555-2287
16840	Adams	John	10.00	Atlanta	GA	404-555-4301
31719	Smith	John	5.50	Detroit	MI	313-555-5186
38102	Baker	George	7.00	Atlanta	GA	404-555-7739
. . .						

شكل 2.2
ملف الرواتب

وإذا استطعنا أن نربط بعض العناصر التي حددناها حتى الآن مع بعض الاصطلاحات العديدة التي تكون جزء من مصطلحات الملفات وقاعدة البيانات فإن ذلك يبسط الأمور فيا بعد. يسمى تجميع الحقائق الخاص بعامل معين (سطر واحد أو صف واحد من الجدول الموجود في شكل 2.2) بسجل record وكل نوع من أنواع الحقيقة (مثل رقم العامل أو المدينة التي يسكن فيها) يسمى حقل field وعلى هذا فيسمى محتوى الجدول الموجود في شكل 2.2 وهو تجميع الحقائق الخاصة بكل العاملين بملف file. الملف الموجود في شكل 2.2 وهو من أكثر أنواع الملفات الأساسية تخيلاً يسمى في العادة بملف بسيط أو خطي Simple or linear file

(خطى حيث أنه تجميع لسجلات ترد واحداً تلو الآخر وذلك في خط طويل).

وسوف نتكلم عن تخطيط كل سجل من سجلات الملف (عناوين الأعمدة في شكل 2.2) كنوع بيانات data type وكل سجل فردي (كل صف في شكل 2.2) كحدوث occurrence لهذا السجل (أو لنوع السجل).

وبوضع ذلك بطريقة مختلفة لكنها مرتبطة لهذه الطريقة يسمى كل شيء نهتم بتتبعه (الشيء الطبيعي الواقعي أو الحدث وليس الحقائق الخاصة به) بكينونة (مفردة) entity. وعلى هذا فإن John Smith الحقيقي الحي الذي يتنفس ويمكنك أن تلمسه هو في الواقع كينونة. وتجميع الكينونات من نفس النوع (مثل كل العاملين في الشركة) يسمى بفئة الكينونة entity set. والخاصية attribute هي صفة أو سمه للكينونة. رقم هاتف Smith ورقم العامل والمدينة التي يسكن فيها وما إلى ذلك كلها خواص له. وبربط هذه المصطلحات بالملفات يمكننا أن نرى السجل يصف describes كينونة محددة (على الأقل بالنسبة لأحد التطبيقات أو للعديد من التطبيقات) ويصف الملف محتويات فئة كينونة. وتمثل الخاصية المعطاة كقيمة حقل.

وإحدى سمات الملف الأخرى في شكل 2.2 خاصة بحقل أرقام العاملين هي أن قيمها فردية. حيث أن كل عامل له رقم فردي فيمكننا أن نستخدم هذا الحقل في تمييزه من بين السجلات المختلفة للعاملين مثل هذا الحقل (أو مجموعة الحقول) يسمى حقل رئيسي (مفتاح) Key للملف.

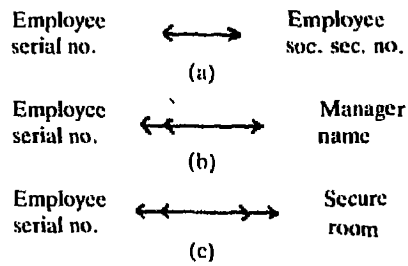
العلاقات

حتى الآن نهتم بحفظ الحقائق عن كينونة محددة. إحدى أنواع المعلومات الأخرى المهمة جداً هي تجميع الطرق التي ترتبط بها الكينونات المختلفة مع بعضها البعض relate to each other. مثل هذا التداخل يوصف كمصاحبات associations وعلاقات relationships.

افرض أنه بالإضافة الى العامل هناك حاجة الى تخزين بيانات عن رقم الضمان الاجتماعي لكل العاملين غير الإداريين والمديريين والغرف التي يوجد فيها العاملون. يوجد هناك رقم ضمان اجتماعي واحد بالضبط يصاحبه مدير واحد لكل

عامل وهذه الحقائق تسمى مصاحبات فردية Unary associations. بالإضافة إلى وجود رقم ضمان اجتماعي واحد يصاحب أحد العاملين فبالنظر إليه من ناحية أخرى فلكل رقم ضمان اجتماعي يوجد عامل واحد بالضبط مصاحب له.

وفي هذه الحالة يكون هناك مصاحبة فردية في كل من الاتجاهين ويسمى ذلك بعلاقة من واحد إلى واحد one-to-one relationship شكل 2.3. a.



شكل 2.3

العلاقات: (أ) من واحد إلى واحد
(ب) من واحد إلى العديد
(جـ) من العديد إلى العديد

بالعودة مرة أخرى إلى العاملين والمديرين فبينما يوجد مدير واحد فقط لكل عامل فإنه لكل مدير يمكن أن يوجد العديد من العاملين الذين يعملون له. وعلى هذا فهناك مصاحبة فردية في اتجاه واحد ومصاحبة متعددة multiple association في الاتجاه الآخر ويسمى ذلك بعلاقة من واحد إلى العديد one-to-many relation ship شكل 2.3. b. (لاحظ أن الأسهم الفردية تمثل المصاحبة الفردية بينما الأسهم المزدوجة تمثل المصاحبة المتعددة.

أخيراً اعتبر العاملين والغرف الآمنة. يمكن أن يسمح لأحد العاملين بالتواجد في العديد من الغرف بينما يمكن أن يسمح للعديد من العاملين بالدخول في غرفة واحدة. ويسمى هذا بعلاقة العديد إلى العديد many - to - many relationship شكل 2.3. b.

الملفات البسيطة

دعنا نسترجع ما سبق أن حددناه حتى الآن. فكل ذلك يبدو بسيطاً لكنه يشكل قاعدة صلبة لما يأتي بعد.

الفكرة العامة الأساسية للبيانات هي إعداد تجميع للحقائق المطلوبة عن الأشياء (الكينونات) الموجودة في بيئتنا التي نهتم بها. وفي المفهوم على الأقل يمكننا

أن نتخيل مثل هذه التجميع لعمل صف من صفوف جدول (سجل من سجلات ملف) والذي يجمع البيانات مع بعضها لكل حدوث لهذا النوع من أنواع الكينونة. مرة أخرى يحتوي الجدول أو الملف الموجود في شكل 2.2 على صف بيانات خاص بكل عامل من العاملين بالشركة.

ويحضرنا ذلك إلى أول مرة من المرات العديدة في هذا الكتاب التي نواجه فيها السؤال التالي:

إن آخر مقطع يبدو كاملاً منطقياً لي كإنسان ولكن كيف لكمبيوتر من مسامير وصواميل (أو بدقة أكثر مكون من دوائر ورقائق) أن يشعر بذلك؟

عند كل نقطة اتصال يجيب هذا الكتاب على مثل هذه الأسئلة على مستوى مناسب لممارس تشغيل البيانات أو للطالب أو للمستفيد وللمدير. ولن ندخل في تفاصيل الهندسة الكهربائية أو الميكانيكية لذلك.

ولكي نعالج أول سؤال يكون من الضروري القاء نظرة على أنواع تمثيلات البيانات المناسبة ووحدات تخزين البيانات المتاحة. وهذه معلومات في واقع الأمر مفيدة جداً خلال هذا الكتاب. خواص هذه الوحدات هي اعتبارات هامة في كل من طرق الاتصال بالبيانات والأداء المتوقع تحقيقه في عمل هذا الاتصال.

مفاهيم تخزين البيانات Data Storage Concepts

تمثيلات البيانات

قد تكون معتاد على مفاهيم البت والبايت والطرق الأساسية لتشكيل البيانات بها ولكن لاتمام هذا الأمر دعنا نبدأ بعرض سريع لهذه المادة.

البت bit أو الرقم الثنائي هو في واقع الأمر مفتاح له اتجاهين ويكون موضوعاً في أي لحظة في اتجاه واحد من هذين الاتجاهين. وعبر السنوات ادمجت هذه الفكرة داخل حلقات معدنية صغيرة (أو قلوب) تحفظ في اتجاه واحد أو في الاتجاه العكسي له أو بوجود بقع على البطاقة المثقبة أو حذف هذه البقع بتثقيبها أو بوجود بقع على أكسيد معدني يغطي مادة البلاستيك أو الألومنيوم ويتم مغنطتها في اتجاه واحد أو في اتجاه آخر وهكذا. وحيث أنه لا يوجد المزيد عن الطريقة التي

يمكن أن تسجل بها البيانات في بت فردية فتجمع مجموعة البت مع بعضها مكونة وسطاً لتخزين البيانات. وفي العديد من أجهزة الكمبيوتر الموجودة حالياً تجمع ثمانية بت مع بعضها مكونة بايت byte واحد. ومجموعات البايت، أو في بعض أجهزة الكمبيوتر مجموعات البت الأخرى، تدمج مع بعضها لتكون كلمات words. والنوعين الأساسيين للبيانات التي تهمنا في التخزين هما البيانات العددية والبيانات الحرفية.

وهناك ثلاثة طرق معتادة في تخزين البيانات العددية وعادة ما تسمى بالثنائي والعشري والنقطة المتحركة. فالعدد الثنائي هو ببساطة عدد ممثل تمثيلاً ثنائياً أو ممثلاً للأساس 2 على هيئة سلسلة من البت كل منها له القيمة صفر أو القيمة 1 كما يتطلب النظام الثنائي ذلك. وعلى سبيل المثال فإن سلسلة البت التالية 0111001 تمثل العدد 57.

العدد العشرين يتكون بتحديد تسلسل بت محدد في بايت ليكون له القيمة صفر وتسلسل آخر يكون له القيمة 1 وهكذا حتى الرقم 9. وعلى سبيل المثال إذا كان 11110101 يمثل الرقم العشري 5 (لاحظ أن آخر 4 بت يكونوا الرقم 5 في التمثيل الثنائي) وإذا كان 11110111 يمثل العدد العشري 7 فإن 11110101 11110111 يمثل 57 في هذا النظام. وحيث أنه مطلوب 4 بت فقط لتمثيل عشرة أرقام عشرية مختلفة فيمكن حفظ هذا التمثيل بحيث أن الرقمين العشريين يظهران في بايت واحد. 01010111

في هذا المثال. ويعرف هذا النظام بالنظام العشري المضغوط Packed decimal وهو مباشر جداً فيما عدا أنه يكون مطلوب بايت إضافي لتمثيل إشارة العدد. والنظام العشري موجه لتطبيقات الأعمال التي لها متطلبات كبيرة من بيانات المدخلات والمخرجات وذلك نظراً لسهولته النسبية في نقل البيانات الى ومن وحدات المخرجات ووحدات المدخلات.

وأعداد النقطة المتحركة موجهة للتطبيقات العلمية والتي عادة ما تتطلب أعداداً كبيرة جداً أو صغيرة جداً. وبينما نجد اختلافات في الموضوع إلا أن عدد النقطة المتحركة يحتوي على ثلاثة أجزاء: الإشارة والخاصية والكسر. والخاصية هي بصفة عامة أس يرفع أساس نظام العد إلى قوة معينة. وعلى هذا تضرب

النتيجة في الكسر للحصول على القيمة المطلوبة. مثال ذلك 243E5 والتي يمكن أن تكون ترميزاً لـ 10 (أساس نظام العد) مرفوع للقوة 5 مضروباً في 243 أو أنه 24,300، فإذا كان موقع التخزين المحدد للكسر صغيراً فإن مدى الأس يسمح بأعداد كبيرة جداً وأعداد صغيرة جداً وقد تفقد بعض الدقة إلا أن هذا لا يؤثر في واقع الأمر على الأعداد والتطبيقات التي تستخدم هذا التمثيل.

وتمثل الرموز كتسلسل بت يحدد بصورة معينة في الشفرة المستخدمة والتي يعمل على أساسها الكمبيوتر. وأكثر شفرتين شائعتي الاستخدام هما الشفرة الثنائية الموسعة للتبادل العشري Extended Binary Coded Decimal Interchange Code (EBCDIC) والتي تمثل كل رمز على أنه تسلسل من ثمانية بت (رمز في كل بايت) والشفرة الأمريكية القياسية لتبادل المعلومات. American Standard Code for Information Interchange (ASCII) والتي تستخدم سبعة بت في التمثيل. مثال ذلك يمثل تسلسل البت 11000001 الرمز A في شفرة EBCDIC ويمثل تسلسل ا لبت 11000010 الرمز B في نفس الشفرة وهكذا. ولتكوين كلمة SMITH في ذاكرة الكمبيوتر يكون هناك حاجة إلى خمسة بايت في التخزين ويحتوي البايت الأول على نمط البت الخاص بالحرف S والثاني على النمط الخاص بالحرف M وهكذا.

وعلى هذا فإن تخزين السجل في ذاكرة الكمبيوتر عبارة عن برنامج يتسبب في إحداث انماط البت الصحيحة للحقول الحرفية من السجل وأحداث التمثيل الصحيح للحقول العددية من السجل وتخزين كل منها في موقع التخزين المتتالي في الذاكرة. والسجل الموجود في شكل 2.1 يمكن أن يخزن في العديد من البايت ليمثل رقم العامل 31719 في صورة عشرية مضغوطة على سبيل المثال يتبعها بايت بنمط بت للحرف S يتبعه بايت بنمط بت الحرف M وهكذا. وبالنسبة لمحتويات الملف يمكن أن تتبع السجلات بعضها البعض في الذاكرة وهذا يحضرنا للنقطة الخاصة بمناقشة الأنواع المختلفة للذاكرات وبعض أنواع وحدات التخزين المختلفة الموجودة في أيامنا هذه.

ويوجد أساساً مستويين من التخزين أو الذاكرات في نظام الكمبيوتر: الذاكرة الابتدائية والذاكرة الثانوية.

الذاكرة الابتدائية

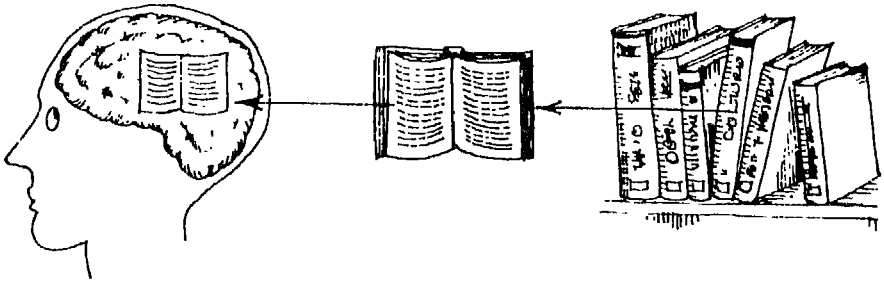
الذاكرة الابتدائية هي المكان الذي يجري فيه تشغيل على البيانات ويوجد به البرامج التي تنفذ وجزء من النظام والذي ينسق ويتحكم في البرامج (نظام التشغيل ونظام إدارة قاعدة البيانات وبرنامج التحكم في الشبكة). ويمكن على مستوى البايت أو مستوى الكلمة تحديد موقع تخزين فردي والوصول إليه مباشرة عن طريق وحدة التشغيل المركزية للكمبيوتر. ونظراً لأن الذاكرة الابتدائية حرجة بهذا الشكل لعمل النظام كله فيجب أن تكون سريعة جداً بالنسبة لتخزين محتوياتها واسترجاعها. ونظرياً يكون من الممتع أن تكون الذاكرة الابتدائية غير محدودة الحجم ويمكن أن يخزن فيها كل البرامج وكل ملفات البيانات المصاحبة للنظام ولكن نظرا للسرعة وللمتطلبات الأخرى على الذاكرة الابتدائية فتتطلب الذاكرة الابتدائية تقنيات مرتفعة التكاليف نسبياً. هذا بالإضافة إلى حقيقة أن الذاكرة الابتدائية محدودة الحجم طبقاً لمخططات عناوين النظام المستخدمة في الاتصال بمحتوياتها يسمح لها بأن تكون كبيرة بدرجة كافية لتخزين جزء صغير جداً من برامج النظام وملفات البيانات في نفس الوقت.

ومن المعتاد أن تتكون الذاكرة الابتدائية من قطع صغيرة على شكل كعكة من الحرير تسمى بالقلوب Cores (يستخدم اصطلاح ذاكرة القلوب كمرادف لاصطلاح الذاكرة الابتدائية) وهي متشابكة مع بعضها عن طريق اسلاك تمر خلالها. ويمثل كل قلب بت واحدة ويمكن للأسلاك التي تمر خلالها أن تحدد كل من اتجاهي المغنطة على القلوب والتي تمثل قيم 0 و 1 ويمكن الإحساس بها لتحديد القيم الحالية في أي وقت لاحق. وحالياً تعد الذاكرات الأولية بواسطة تقنيات الدوائر المتكاملة الأحدث. ويبقى مفهوم البت كما هو إلا أن السرعات المشمولة ازدادت بطريقة هائلة.

وفي أي حالة من الحالات يمكن التفكير في الذاكرة الابتدائية كمنظومة هائلة من البت مرتبة على هيئة بايت أو كلمات. وكل كلمة أو بايت لها عنوان ويمكن الاتصال بها عن طريق وحده التشغيل المركزية وذلك لتخزين بيانات جديدة أو لاسترجاع بيانات موجودة وذلك بسرعة عالية.

الذاكرة الثانوية

الحاجة إلى ذاكرة ثانوية: نظراً لندرة الذاكرة الابتدائية فتحفظ الأحجام الكبيرة من البيانات وكذلك البرامج التي تجري تشغيل عليها في وحدات الذاكرة الثانوية. وتحمل السجلات الفردية أو مجموعات السجلات الموجودة في الملف من الذاكرة الثانوية داخل الذاكرة الابتدائية عندما يراد اجراء التشغيل عليها (كما يحدث ذلك مع البرامج عند تنفيذها أيضاً). ويمكن عمل تماثل بين الذاكرة الابتدائية والذاكرة الثانوية في الكمبيوتر مع مخ الإنسان والمكتبة شكل 2.4. لا يستطيع المخ أن يحفظ كل المعلومات التي يحتاج إليها الإنسان ولكن دعنا نقول أن المكتبة الكبيرة يمكن أن تفعل ذلك وعندما يريد الإنسان معلومات محددة ليست موجودة في ذاكرته أو في مخه في هذه اللحظة فإنه يجد كتاباً في المكتبة يحتوي على هذه المعلومات وبقراءة هذا الكتاب يقوم بنقل المعلومات من الكتاب إلى مخه.



شكل 2.4
الذاكرة الابتدائية والذاكرة الثانوية يشبهان المخ والمكتبة

الاتصال التتابعي والاتصال المباشر: بينما تأتي تطبيقات تشغيل البيانات في عدد غير محدود من الصيغ إلا أنه هناك نوعين أساسيين من أنواع الاتصال بالبيانات المخزنة لأي تطبيق من هذه التطبيقات المختلفة. وسوف نشير في هذا الكتاب إلى هذين النوعين من الاتصال على انهما اتصال تتابعي واتصال مباشر.

ويعني اضطلاح الاتصال التتابعي Sequential access انه يجب أن يسترجع التطبيق ويشغل كل سجلات الملف أو جزء منها واحداً تلو الآخر حتى ينتهي تشغيل كل السجلات المطلوبة. ويمكن أن ترتب سجلات الملف طبقاً لقيم بعض

الحقـول الرئيسيـة ويمكن أن تكون في تـرتيب طبقـاً لـوصـول السجـلات بحيث أن أحدث سجل تم إدخاله في الملف يكون آخر سجل في الملف. كما يمكن ألا تكون هذه السجلات في أي ترتيب على الإطلاق. وهناك مثال شائـع للتشغيل التتـابعي وهو انتاج شيكـات رواتب إحدى الشركـات أسبوعيـاً. حيث أن كل العـاملين في الشركة يحصلون على رواتبهم فيجب المـرور على كـل سجلات العـاملين الخاصـة بـالـرواتب والتي يمكن أن تـرتب طبقـاً لحقـل رقم العـامـل ويحـدث المـرور عـلى السجـلات سجلاً سجـلاً حتى ينتهي إعداد شيكـات الرواتب لجميـع العـاملين. وتـرتيب الملف طبقاً لـرقم العامـل يسهـل من التحكم في الملف وتجـديـده ويبسط كذلك من عملية توزيع الشيكات.

طـريقة الاتصـال الأخرى يشـار إليها في هـذا الكتاب بـأنها اتصـال مبـاشر direct access. وتوصف عند الحاجة إلى إيجـاد سجل محـدد في ملف واسترجـاعه بالافتراض المعتاد ولكن غير المقيـد بأنـه يجب استعادتـه بسرعة كبـيرة. ويسترجـع السجل أما لتقديمه ببساطة لشخص آخر لاستخدامـه أو لتشغيله أو تغييره بـطريقة معينة أو لكلا الفرضين. ومثال هذا النوع من الاتصال بـالبيانـات هو نـظام حجز مقـاعد الـطائرة. فعنـدما يتصـل العميل هـاتفياً بشركـة الطيران لعمـل حجـز أو للحصـول على معلومـات عن رحلة معينة في يـوم معين فيجب أن يكـون الموظف والذي عادة ما يكون جالساً أمام نهاية طرفيه بها شاشة قادراً على استرجاع بيانـات الرحلة المطلوبة بسرعة. ويجب أن ندفع مثـال ملف رواتب العاملين لـلإلمام بعض الشيء في هذا الاتجاه عن طريق اقتراح أنـه يكون هنـاك أوقات يجب أن يسـترجع فيهـا سجل رواتب أحـد العاملين بسرعـة أيضاً. مثـال ذلك إذا مـا حصـل أحـد العاملين على زيادة في الـراتب ونريد أن نسجل هذه البيانات الجـديدة فنكـون في حاجة الى إيجاد وتشغيل سجل رواتب هذا العامل فقط.

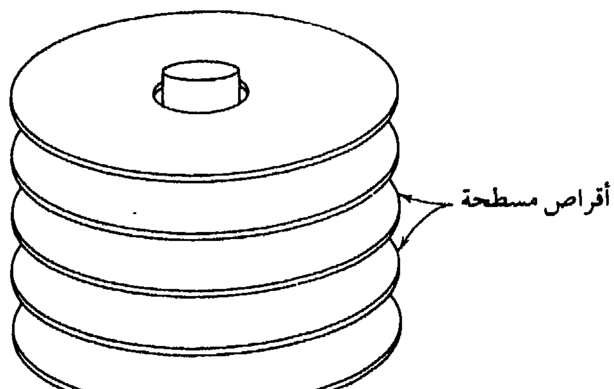
وصيغتي الذاكرة الثانوية الشاملتين في الاستخدام في وقتنا الحالي هما الشريط المغناطيسي والقرص المغناطيسي. الشريط المغناطيسي هو وسط تخزين تتـابعي وعلى هذا فهو يدعم الاتصال التتابعي وتخزين البيانات رمزاً وراء رمز في طـول الشريط ولا يمكن عمل اتصال مباشر مع الشريط. معظم تطبيقات قواعد البيانات الحاليـة تتطلب اتصالاً مباشراً وعلى هذا فالشريط ليس عنصراً في عالم قاعدة البيانات.

وحدات تخزين الأقراص

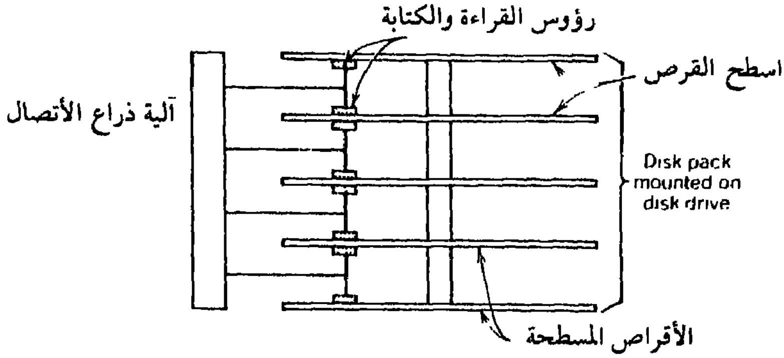
تتطلب قاعدة البيانات وحدات تخزين كبيرة جداً في سعتها التخزينية، وسريعة جداً في أوقات التخزين والاتصال (بالرغم من أنها لا تكون بنفس سرعة التخزين الابتدائي)، وتوفر اتصالاً مباشراً للبيانات، وتقل تكلفتها عن وحدات الذاكرة الابتدائية لنفس السعة ومن مثل هذه الوحدات شائعة الاستخدام حالياً وحده القرص المغناطيسي magnetic tape.

ويكون القرص المغناطيسي الفردي مسطحاً من الألومونيوم الصلب الدائري المغطى بأكسيد الحديد على سطحية. وعادة ترص عدة أقراص مع بعضها وتركب على عمود مركزي مع وجود بعض الفراغات بينها وذلك في مجموعة أقراص disk pack، شكل 2.5. ويكون ترتيب مجموعة الأقراص مستديماً، فلا يمكن تبادل أقراص المجموعة أو إزالتها. وتوضع مجموعة الأقراص في وحدة تعرف بأنها مشغل قرص disk drive يكون له رؤوس قراءة وكتابة قادرة على تخزين بيانات واسترجاعها على أسطح القرص ومنها، شكل 2.6. وتدور مجموعة الأقراص على مشغل الأقراص بسرعة مرتفعة (مثل 2400 دورة في الدقيقة) متسببه في جعل أي قطعة بيانات معينة قابلة للاتصال بها العديد من المرات كل ثانية.

ومن المهم فهم كيف ترتب البيانات على الأقراص وعوامل الأداء المرتبطة بذلك من عمليات الاسترجاع. تخزن البيانات بت بعد بت تتابعياً في دوائر لها نفس المركز على القرص تعرف بالمسارات tracks، شكل 2.7. ويمكن أن يوجد



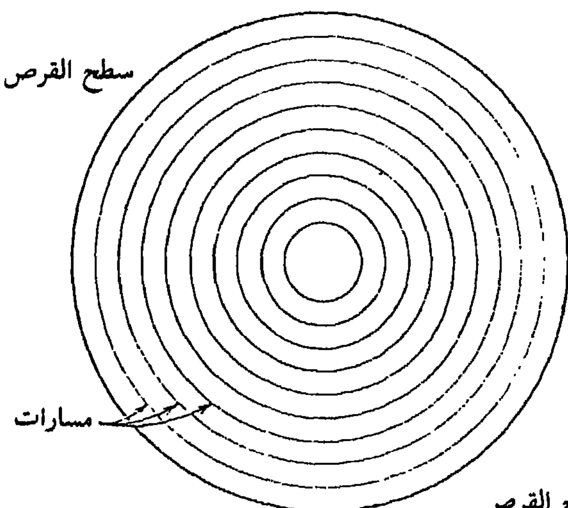
شكل 2.5 : **مجموعة أقراص**



شكل 2.6 : مجموعة أقراص مركبة تبين آلية ذراع الاتصال مع رؤوس القراءة والكتابة

أقل من 100 أو عدة مئات من المسارات على كل سطح من أسطح القرص، اعتماداً على الوحدة نفسها. دعنا نعطي آخر مسار على القرص من الخارج الرقم 0، والمسار التالي له للداخل رقم 1، وهكذا. يمكنك أن تفترض أن طريقة تحميل القرص من مجموعة أقراص يمكن أن تكون عن طريق ملأ كل مسارات أحد أسطح القرص، ثم ملأ مسارات السطح التالي له، وهكذا. وفي واقع الأمر ليست هذه هي الطريقة المتبعة في الملأ.

المسار رقم 0 على السطح رقم 0، والمسار رقم 0 على السطح رقم 1، وكل مسار له رقم 0 على كل الأسطح المتبقية للمجموعة تدمج منطقياً في الأسطوانة رقم



شكل 2.7 : مسارات على سطح القرص

(Cylinder 0). ونفس ترتيب الأسطوانة المنطقي يعد لكل مجموعة من المسارات التي لها نفس رقم الفهرس على أسطحها المناظرة. هذا ومثلما تنظر إلى مجموعة الأقراص من أعلى وتختار مجموعة من المسارات الدائرية التي لها نفس المركز على أن تأخذ مساراً من كل سطح وكل المسارات التي تقع فوق بعضها تماماً تشكل اسطوانة. وعلى هذا فإن إجمالي عدد الإسطوانات في مجموعة الأقراص يساوي عدد المسارات الموجودة على أي سطح من أسطح أقراص المجموعة.

وسبب الاهتمام بمفهوم الاسطوانة أهميته بالنسبة لنظم مكونات القراءة والكتابة في مشغل القرص. فقد يكون من المكلف جداً والمرهق جداً وجود رأس قراءة وكتابة لكل مسار من كل سطح. وعلى هذا فإن الترتيب الموجود في شكل 2.6 هو من أقدم وأبسط هذه الأنواع وتوجد آلية اتصال متحركة تتكون من مجموعة من الأذرع بها رؤوس القراءة والكتابة. وكل ذراع به رأسين للقراءة والكتابة في نهايته وموجود في الفراغ الموجود بين قرصين. في هذا التصميم تكون إحدى رؤوس القراءة والكتابة على الذراع قادرة على الاتصال بسطح القرص الذي يقع فوق الذراع والرأس الأخرى تكون قادرة على الاتصال بالسطح الذي يقع أسفل الذراع. وعلى هذا فكما يمكن رؤيته في شكل 2.6 فهناك رأس قراءة وكتابة لكل سطح. وأكثر من ذلك فإن هذا المحتوى الذي يشبه المشط لتجميع أذرع الاتصال مصمم لينقل كجزء واحد متماسك مع بعضه. فلا يمكنك أن ترى رأس السطح 0 موضوعه على المسار 10 لهذا السطح وفي نفس الوقت تكون رأس السطح 3 موضوعة عند المسار 35 لهذا السطح. فإذا كانت إحدى الرؤوس عند المسار 10 للسطح الخاص بها فإن جميع الرؤوس تكون عند المسار 10 كل منها للسطح الخاص بها وهذا الترتيب له شيئين يتبعانه. أحدهما هو أنه يقلل تكاليف ودرجة تعقيد مشغلات الأقراص. والآخر هو أنه ينتج مفهوم الإسطوانة المنطقية لأسباب موجهة للأداء التالي.

هناك أربعة خطوات أساسية أو اعتبارات وقتية في نقل البيانات من مجموعة اقراص إلى التخزين الابتدائي:

١ ـ وقت البحث Seek time نقل ذراع الاتصال الى الاسطوانة الصحيحة من أي موقع يكون موجود فيه فعلاً.

٢ ـ اختيار عمل الرأس head switching اختيار القراءة أو الكتابة للرأس التي

تصل إلى مسار محدد (أو سطح محدد) . من الاسطوانة الموجود فيها البيانات .

٣ ـ زمن تأخير الدوران rotational delay انتظار البيانات المطلوبة حتى تصل تحت رأس القراءة والكتابة على دوران القرص .

٤ ـ وقت النقل transfer time الوقت الفعلي اللازم لنقل البيانات من القرص الى الذاكرة الابتدائية بمجرد إتمام الخطوات من 1 إلى 3 .

لا يأخذ النقل لرأس الكتابة والقراءة للكتابة أو للقراءة والعكس وقتاً في الواقع حيث أن الأمر لا يزيد عن تغيير حالة مفتاح كهربائي ويأخذ وقت تأخير الدوران في المتوسط حوالي نصف دورة من دورات القرص الكاملة ولا يمكن تجنبه على أية حال. وقت النقل هو وقت سريع جداً وهو مرة أخرى، ضروري في أي حالة من الأحوال. وأبطأ جزء في العملية ولا يمكن عمل شيء بالنسبة له هو وقت البحث : حركة ذراع الاتصال.ويكون هناك طريقة لتحويل الموقف هذا الى ميزة مع وجود مفهوم الإسطوانة. إذا كانت سجلات أحد الملفات مرتبة طبقاً لحقل رئيسي معين فيكون من الافتراضات الجيدة افتراض أنه بالنسبة لبعض التطبيقات على الأقل التي تستخدم الملف يتم الاتصال بالسجلات واحداً تلو الآخر طبقاً لنفس الترتيب الرئيسي. والأكثر من ذلك ولأسباب سوف نتعرض لها فيما بعد في هذا الكتاب فهناك مواقف قواعد بيانات عديدة والتي تخزن فيها اجزاء البيانات المرتبطة ببعضها في أشكال سجلات مختلفة يتم الاتصال بها مع بعضها. وفي كل من هاتين الحالتين إذا ما خزنا البيانات منذ البداية عن طريق ملء كل مسارات إحدى الاسطوانات فيمكن بعد ذلك استرجاع كل هذه البيانات دون الحاجة إلى نقل ذراع الاتصال.

نضع مجموعة أذرع الاتصال عند الاسطوانة الصحيحة ونبدأ في القراءة من أول مسار مطلوب من هذه الاسطوانة ثم يغير فقط من مفتاح اليكتروني لتنشيط رأس أخرى للقراءة والكتابة وقراءة مسار آخر من نفس الاسطوانة. البديل هو ملء كل مسارات أحد أسطح أحد الأقراص ثم الانتقال إلى مسارات سطح أخرى والذي يتطلب دائماً حركة مجموعة أذرع الاتصال لتذهب من البيانات الموجودة على مسار قريب منها (على نفس سطح القرص). وعلى مدار السنوات

كان هناك تحسينات هائلة وتغييرات واسعة في تقنية الاقراص وتشمل ما يلي:

* مدى من 1 إلى ما يصل إلى 16 قرص في مجموعة الأقـراص (وتقدر هـذه من 2 إلى 30 مسار في كل اسطوانة حيث أنه بالنسبة لمجموعات الأقراص لا يتسخدم في العادة كل من السطح العلوي والسطح السفلي للمجموعة).

* صممت بعض مشغـلات الأقـراص بحيث يمكن نقـل مجمـوعـات الأقـراص واستبدال المجموعـة بمجموعـة أخرى كلما كـان هناك رغبـة في ذلك وصممت بعضهـا بحيث تكون الأقـراص ثابتـة بصفة دائمـة. وميزة مجمـوعة الأقـراص الثـابتة هي أن المسـاحات الهنـدسية يمكن أن تكـون أدق مـع السـماح بكثـافـة تسجيل أعلى للبيانات.

* في بعض الحالات تكون مجموعة أذرع رؤوس القراءة والكتابة جزءاً من وحـدة مشغل الأقراص وفي حالات أخرى تكـون جزءاً من مجمـوعة الأقـراص نفسها وتتصل بالمشغل عندما يتم تركيب مجموعة الأقراص عليه.

* وعـلى عكس التصميم الأصلي لمجمـوعـة أذرع اتصـال واحـدة تصـل إلى كـل البيانات الموجودة على مجموعة الأقراص فهناك تصميمات أكـثر حداثـة بها أكـثر من مجمـوعة أذرع اتصـال واحدة. وفي بعض الحـالات تصل مجمـوعات أذرع مختلفة بأسـطح أقراص مختلفـة وفي حالات أخـرى فإنها تصـل إلى اسطوانـات مختلفـة لنفس مجموعـة الأسطح. ومن الـواضـح أن هـذه الاختـلافـات تفتـح الطريق للاتصال المتعدد والمتوازي بالنسبة لمجموعة أقراص معينة.

* حديثاً وفي مجال الميكروكمبيوتر بصفة خاصة وجدت القريصات. وهذا هو وجه من الأوجه الذي يوجد فيه قرص واحد (سطحـين) وقطرة لا يـزيد عن بضـع بوصات ومصنوع من البلاستيك المغطى بأكسيد الحـديد والقـرص رقيق ومرن وعنـدما يـدار القرص في مشغـل الأقراص فيمكن استخـدامه بنفس الـطريقـة المعتادة لاستخدام الأقراص.

* وهناك نوع شبيه آخر من الواحدات يسمى بالاسطوانات المغناطيسيـة magnetic drums وهي وحـدة اسطوانيـة يـوجـد عـلى سـطحهـا مسـارات دائـريـة. والاسطوانة تكون مثبتة تثبيتاً دائماً في الوحدة التي تتسبب في دورانها ويوجد بها

رأس قراءة وكتابة لكل مسار من مساراتها ونتيجة لهذا الترتيب يكون الوصول إلى البيانات سريعاً جداً ولكن نظراً لسعة التخزين المحدودة وارتفاع التكلفة فعادة ما تستخدم الاسطوانات في حفظ برامج النظام والبيانات المصاحبة لها على عكس ملفات بيانات التطبيقات.

تشكيلات البيانات والقرص

هناك أمران آخران كل منهما يهتم بأوجه ترتيب البيانات وسوف نعتبرهما الآن. الأمر الأول يشمل اختلافات في الطريقة التي يمكن ترتيب السجل بها. والآخر يشمل أموراً خاصة بوضع البيانات على القرص.

تشكيلات السجل: تسمى السجلات الموجودة في الملف الموجود في شكل 2.2 بسجلات ثابتة الطول fixed length فكل سجل في الملف له نفس الطول وله نفس أنواع الحقول مثل بقية السجلات الأخرى. والبديل الآخر هو الملفات التي تحتوي على سجلات متغيرة الطول variable length وكل سجل من هذه السجلات يمكن أن يحتوي على حقول مختلفة كما يمكن أن يختلف طول السجل في مدى محدد.

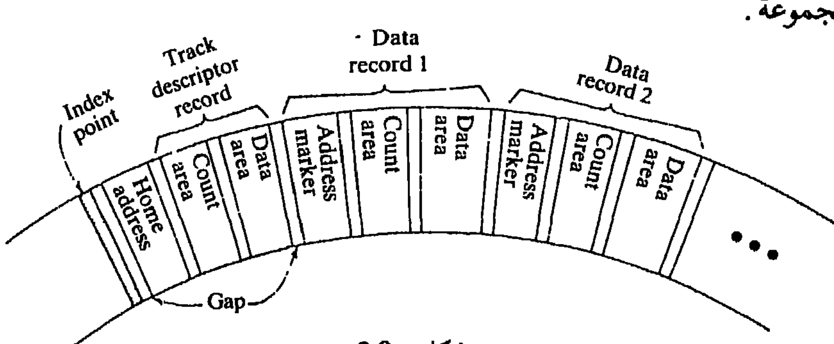
ويسمى كل سجل من سجلات الملف الموجود في شكل 2.2 بسجل منطقي logical record. وحيث أن معدل تشغيل البيانات في وحدة التشغيل المركزية أسرع كثيراً من المعدل الذي يمكن إحضار البيانات به من الذاكرة الثانوية فمن المرغوب فيه نقل العديد من السجلات المخزنة وراء بعضها مرة واحدة. مرة أخرى كما سبق ذكره فإن هذا له تأثيرات على كل من الملفات التتابعية البسيطة وعلى تكوين قاعدة البيانات كذلك وبعد إحضار سجل واقعي physical record أو مجموعة block من العديد من السجلات المنطقية داخل الذاكرة الابتدائية من الذاكرة الثانوية فيمكن أن يفحص كل سجل منطقي ويتم تشغيله بواسطة البرنامج الذي يحتاجه كلما كان ذلك ضرورياً.

تشكيلات المسار: أخيراً يوجد السؤال الخاص بكيفية وضع سجلات البيانات في مسارات القرص بحيث يمكن أن توجد عندما يحدد النظام أنها موجودة على مسار معين.

ومن تشكيلات المسار التقليدية يوجد تشكيل عد البيانات Count - Data

وتشكيل عد المفتاح الرئيسي للبيانات Count - key - data المستخدمين في وحدات تشغيل أقراص IBM 3330 و IBM 2305 (مع بعض التعديلات) ومع وحدات تشغيل أقراص IBM 3340 و 3344 و 3350 و 3375 و 3380. ويوضح شكل 2.8.a تشكيلات عد البيانات. ويحدد بدء المسار بعلامة تعرف بأنها نقطة الفهرس index point يلي ذلك فراغ صغير. وتفصل الفراغات مختلفة الأطوال كل العناصر المختلفة الموجودة على المسار وتعطي وقتاً لمشغل القرص لأداء وظائف معينة. يلي ذلك العنوان المحلي home address والذي يشمل معلومات مثل رقم الاسطوانة ورقم المسار وإشارة تحدد ما إذا كان المسار معيباً أم لا. يلي ذلك سجل خاص يسمى سجل وصف المسار track discriptor record ويحتوي على معلومات مثل عدد البايت المتروك على المسار بعد أخر سجل بيانات وما إذا كان هذا المسار به عيوب أم لا وعنوان المسار البديل الذي يحتوي على بيانات يجب أن تكون على هذا المسار. وأخيراً نصل إلى سجلات البيانات والتي تحتوي على ثلاثة أجزاء: علامة عنوان address marker ومنطقة عد Count area ومنطقة بيانات data area. وعلامة العنوان هي مؤشر خاص يسمح للمشغل بتحديد بداية السجل عند حدوث استرجاعات متتالية. وتكرر منطقة العد أرقام الاسطوانة والمسار وتحديد ما إذا كان المسار معيباً أم لا. وتحتوي على معلومات إضافية مثل رقم السجل (من أرقام السجلات الموجودة على المسار) وطول منطقة البيانات التي تلي ذلك وحقيقة أن هذا الملف موجود في تشكيل عد بيانات على عكس تشكيل عد المفتاح الرئيسي للبيانات.

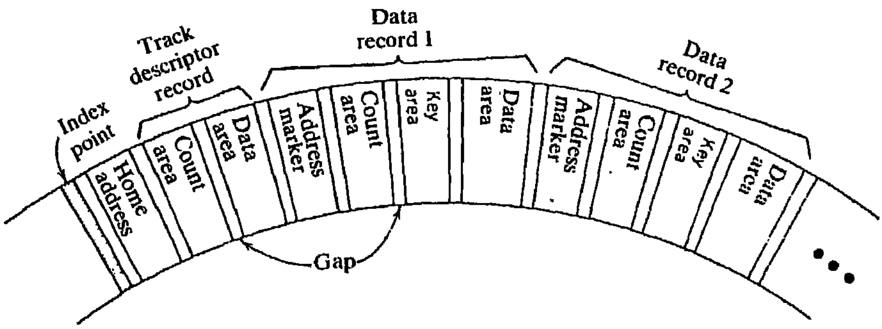
وتحتوي منطقة البيانات على كل البيانات الفعلية للسجل الواقعي أو للمجموعة.



شكل 2.8.a

تشكيل مسار قرص IBM: تشكيل عد البيانات

وتشكيل عد المفتاح الرئيسي للبيانات، شكل 2.8.b، يشبه تشكيل عد البيانات فيما عدى أنه هناك منطقة مفتاح key area بين منطقة العد ومنطقة البيانات وذلك لكل سجل من سجلات البيانات. ويوجد كذلك مؤشر في منطقة العد بطول المفتاح. وعادة ما تحتوي منطقة المفتاح على قيمة لأعلى مفتاح من قيم مفتاح السجلات المنطقية الموجودة في المجموعة. وبسبب مناطق المفتاح هذه يمكن استخدام تشكيل عد المفتاح الرئيسي للبيانات بواسطة النظام لتوقيع السجلات على أساس قيمة المفتاح بطريقة أسرع كثيراً.



شكل 2.8.b
تشكيل مسار قرص IBM: تشكيل عد المفتاح الرئيسي للبيانات

وهناك تطوير أكثر حداثة وهو معمارية المجموعة الثابتة fixed - block architecture وتستخدم في مشغلات أقراص IBM 3310 و IBM 3370. وبينما يقوم المستفيد في مشغلات الأقراص التي سبق ذكرها بأعداد طول المجموعة (معامل المجموعة blocking factor هو عدد السجلات المنطقية الموجودة في سجل واقعي واحد) وذلك بالنسبة لملف بيانات معين فإن وحدات معمارية المجموعة الثابتة FBA يكون لها قيمة ثابتة وهي 512 بايت لكل مجموعة لجميع ملفات البيانات. وتضع معالم نظم مكونات ونظم برامج خاصة البيانات بطرق فريدة في وحدات معمارية المجموعة الثابتة FBA.

تخزين الملفات البسيطة Storing Simple Files

دعنا نعود الآن الى سؤالنا السابق عن كيفية أخذ مفهوم الملف البسيط وتحويله إلى صورة مناسبة لتخزين الكمبيوتر واستخدامه.

كما سبق أن رأينا في هذا الفصل يمكن تمثيل كل من البيانات العددية والبيانات الحرفية كسلسلة من البت. وأي سجل هو ببساطة عبارة عن سلسلة مركبة من كل البت اللازمة لتمثيل كل حقوله. والملف عبارة عن سلسلة مركبة من كل البت اللازمة لتمثيل كل سجلاته.

ويؤخذ قرار بعدد السجلات المنطقية التي يجب أن توجد في سجل واقعي واحد. وبعد ذلك وباستخدام تشكيل عد المفتاح الرئيسي للبيانات يحمل الملف سجلاً واقعياً يلي سجلاً واقعياً آخراً وذلك في مجموعة الأقراص. وتحمل البيانات على التوالي طبقاً للبت على أحد مسارات الاسطوانة يلي ذلك المسار التالي من مسارات الإسطوانة وهكذا حتى تمتلىء. وتستمر العملية مع الإسطوانة التالية فالتالية حتى يتم تحميل كل محتويات الملف.

مراجع References

Bohl, M. Introduction to IBM Direct Access Storage Devices, Science Research Associates, Chicago, IL, 1981.

Martin, J., Computer Data - Base Organization, 2nd Edition, Prentice -Hall, Englewood Cliffs, NJ, 1977.

أسئلة وتمارين Questions and Exercises

١ - ما البيانات؟ هل تعتقد أن كلمة بيانات يجب أن تعامل على أنها كلمة مفردة أو جمع (data)؟ لماذا؟

٢ - حدد أسماء بعض الكينونات والخواص في بيئة الجامعة.

٣ - حدد أسماء بعض الكينونات والخواص التي تهتم بها إحدى شركات التأمين.

٤ - حدد بعض أسماء الكينونات والخواص التي يهتم بها أحد معارض بيع الأثاث.

٥ - ما الفرق بين نوع السجل وحدوث السجل. حدد بعض الأمثلة.

٦ - الملف التالي يستخدم في نظام تشغيل بيانات لأحد منتجي السيارات.

Vehicle Number	Vehicle Model	Color	Date of Manufacture
⋮			
527360	Sedan	Blue	9-12-84
527482	Coupe	Red	9-23-84
630973	Sedan	Red	9-27-84
644182	Sedan	Brown	9-27-84
⋮			

أ ـ حدد أحد أسماء الكينونات الموصوفة في الملف. كيف يمكنك أن تصف فئة الكينونة؟

ب ـ ما هي خواص الكينونات؟ اختر أحد هذه الكينونات وصفه.

جـ ـ اختر أحد الخواص وناقش طبيعة مجموعة القيم التي يمكن أن تأخذها.

٧ ـ اعتبر الكينونات التي تذكرها في الأسئلة 2 و 3 و 4 وصف بعض العلاقات الموجودة بينها.

٨ ـ لماذا تكون العلاقات بين الكينونات مهمة؟

٩ ـ اكتب العدد 84 بالتمثيل الثنائي والعشري والنقطة المتحركة.

١٠ ـناقش أسباب وجود كل من الذاكرات الابتدائية والثانوية في نظام الكمبيوتر.

١١ ـصف أحد التطبيقات الذي يتطلب اتصال تتابعي وتطبيق آخر يتطلب اتصال مباشر وذلك لملف السيارات الموجود في السؤال رقم 6.

١٢ ـ ما الإسطوانة الموجودة في مجموعة الأقراص؟ وما أهميتها بالنسبة لتطبيقات تشغيل البيانات؟

١٣ ـصف بكلمات من عندك كيف تنقل البيانات من مجموعة الأقراص الى الذاكرة الرئيسية.

١٤ ـلماذا يكون من الضروري وجود تشكيلات خاصة للبيانات على القرص؟ صف أحد هذه الترتيبات.

الفصل الثالث
ترتيبات الملفات وطرق الاتصال

File Organizations and Access Methods

الإتصال بالبيانات Accessing Data

لقد تعرضنا في الفصل السابق لمفهوم الملف البسيط مستخدمين ملف الرواتب كمثال. دعنا نتحدث الآن عن الطرق التي يمكننا أن نستخدمها في تطبيق الرواتب وفي التطبيقات المرتبطة به.

هناك أربعة اجراءات يمكن أن نتخذها مع الملف:

(١) النظر ببساطة الى سجل دون تغييره.

(٢) تعديل سجل (مثل اجراء تغيير في معدل راتب أحد العاملين).

(٣) إدخال سجل جديد (مثل تعيين عامل جديد بالشركة).

(٤) حذف سجل موجود بالفعل (مثل ترك أحد العاملين للشركة).

وطبقاً لطبيعة التطبيق الخاصة يمكن للبرنامج المعطي أن ينفذ كل أو بعض هذه الأعمال الأربعة على الملف الذي يصل إليه ولكن بالإضافة إلى ذلك فهناك السؤال عن أي سجل أو أي سجلات نريد تشغيلها في وقت معين.

ومن السهل بدرجة كافية رؤية البيانات في ملف الرواتب وكذلك فهم مفاهيم الاسترجاع التتابعي والاسترجاع المباشر. وقد تحدثنا عن الأسس العامة لوضع البيانات بالفعل على قرص. ولكن ترتيب البيانات على قرص بحيث تكون متاحة للاسترجاع التتابعي أو الاسترجاع المباشر أو كلاهما بطريقة مرتفعة الكفاءة هو مستوى آخر من مستويات التفا صيل والذي يجب التعرض له قبل أن نستطيع أن نستمر. وهو أمر صعب بعض الشيء ولكنه يمكن فهمه بكل تأكيد إذا ما أخذناه بالتدريج خطوة خطوة.

ونشير إلى طريقة تخزين البيانات ليمكن استرجاعها فيها بعد بترتيب الملف file organization. وطريقة استرجاع البيانات المعتمدة على كيفية تخزينها طبقاً لترتيب معين للملف-تسمى طريقة الاتصال access method. (لاحظ أنه بصورة عابرة يستخدم كل من المصطلحين ترتيب الملف file organization وطريقة الاتصال access method كمترادفين إلا أن هذا خطأ من الناحية التقنية).

ترتيب الملف وطرق الاتصال

File Organization and Access Methods

مقدمة:

دعنا نتعرض لسؤال ترتيبات الملفات وطرق الاتصال بنفس بيانات المثال الذي سبق ذكره. تذكر أنه كما يوضح شكل 3.1 يحتوي كل سجل على نفس مجموعة الحقول مثل بقية السجلات الأخرى الموجودة في المجموعة. ونريد أن نحفظ لكل عامل رقم العامل الخاص به والاسم الأخير والاسم الأول ومعدل الراتب وما إلى ذلك. مرة أخرى سوف نشير إلى هذا التخطيط العام أو التكوين العام للسجل بأنه نوع السجل record type وإلى البيانات الخاصة بعامل معين (31719 و Smith و John و 5.50 و...) بأنها حدوث occurence لهذا السجل أو ببساطة بأنها سجل. سوف ننظر إلى الترتيب الرئيسي للملف وإمكانيات طرق الاتصال الموجودة لكل التجميعات البسيطة للسجلات التي تكون كلها من نفس النوع.

Employee Number	Employee Last Name	Employee First Name	Salary Rate	Home city	Home state	Home Telephone
06337	Jones	Fred	4.30	New York	NY	212-555-9207
09155	Smith	Susan	5.90	Atlanta	GA	404-555-2287
16840	Adams	John	10.00	Atlanta	GA	404-555-4301
31719	Smith	John	5.50	Detroit	MI	313-555-5186
38102	Baker	George	7.00	Atlanta	GA	404-555-7739
⋮						

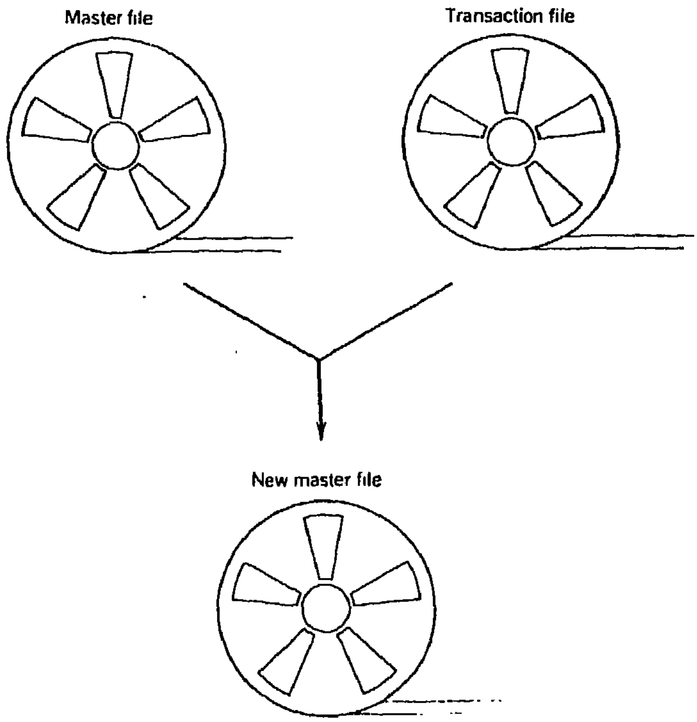
شكل 3.1 ملف الرواتب

الملفات البسيطة الخطية

الملفات التتابعية: أبسط طريقة لتخزين تجميع من السجلات هو بوضعها في قائمة واحدة طويلة. فإذا كانت السجلات متتابعة بالنسبة إلى أحد حقولها أو إلى أكثر من حقل من حقولها فإننا نسمي هذا الترتيب ملفاً تتابعياً sequential (أو ملف مفتاح تتابعي key - sequential). وعلى هذا فملف الرواتب (شكل 3.1) هو في الواقع ملف تتابعي. والحقل المستخدم في عمل تتابع الملف يسمى حقل التتابع sequence field. الحقل الذي له قيم فردية تعرف سجلات الملف يسمى حقل رئيسي key field. وفي ملف الرواتب السجلات مرتبة طبقاً للحقل الرئيسي وعادة، وليس دائماً، يكون الوضع كذلك.

الطريقة الوحيدة لاسترجاع البيانات في الملف التتابعي البسيط هي البدء عند بداية الملف وقراءة سجل سجل على التتابع حتى يتم الوصول إلى السجل المطلوب البحث عنه. ويبدو هذا مخيفاً إذا كنت تبحث عن سجل محدد واحد وهذا فعلاً مخيف. فالملفات التتابعية بدون الوحدات الإضافية التي سوف نناقشها بإيجاز ليست لتطبيقات الاتصال المباشر على الإطلاق. إلا أنها تتسم بالبساطة. وهناك العديد من التطبيقات يناسبها مثل هذه الملفات تماماً. بالعودة الى تطبيق الرواتب نفترض أننا نريد في نهاية كل أسبوع كتابة شيكات الرواتب لكل العاملين في الشركة. لعمل ذلك نحتاج إلى سجل لكل عامل من العاملين في الشركة وذلك في ملف الرواتب. وربما نقرأها كذلك سجلاً وراء الآخر من ملف رواتب كالموضح في شكل 3.1. فإذا كان اهتمامنا بكتابة الشيكات بترتيب آخر غير ترتيب رقم العامل فربما يجب تخزين الملف طبقاً للترتيب الآخر.

لقد وجدت الملفات التتابعية واستخدمت وقتاً طويلاً والطرق الخاصة لتشغيلها هي أجزاء تقليدية من المعرفة بالكمبيوتر. بالإضافة إلى مشكلة الاتصال البسيطة التي سبق مناقشتها هناك مشاكل في إدخال سجلات جديدة وحذف سجلات موجودة بالفعل. وبعد كل شيء فأنت لا تستطيع أن تدخل سجل جديد بين سجلين موجودين بالفعل في وحدة التخزين ولا يمكنك أن تحذف سجل إلا إذا استطعت أن تجده أولاً الشيء الذي يعيدنا الى مشكلة الاتصال المباشر مرة أخرى. والطريقة التي يؤدي بها ذلك موضحة في شكل 3.2 وذلك لتجديد ملف



شكل 3.2
تجديد ملف تتابعي على شريط

تتابعي (يسمى ملف رئيسي) موجود على شريط. يوضع ملف عمليات جارية transaction file على شريط آخر. ويشمل هذا الملف السجلات الجديدة المراد ادخالها وقيم الحقل الرئيسي للسجلات المراد حذفها (مع إضافة رمز مناسب للحذف) وقيم الحقل الرئيسي للسجلات التي يراد تجديدها بالإضافة الى البيانات الجديدة التي تشمل التجديد. وكل هذه مرتبة مع بعضها بنفس تتابع الحقل الرئيسي المرتب به الملف الرئيسي. ويبدأ البرنامج القراءة خلال الشريطين مع البحث عن المواءمة بين الحقلين الرئيسيين بالنسبة لعمليات الحذف والتجديد والبحث عن النقطة المناسبة لوضع السجلات الجديدة بالنسبة لعمليات إضافة السجلات. وبينما تحدث القراءة خلال ملفي الشريط الرئيسي والعمليات الجارية يتم انتاج ملف الشريط الجديد والمسمى بالملف الرئيسي الجديد new master والذي يشمل كل البيانات الموجودة في الملف الرئيسي القديم والتي أجرى عليها

التغييرات الموجودة في ملف العمليات الجارية. والسجلات التي لا تتأثر تنقل ببساطة على شريط الملف الرئيسي الجديد أما السجلات المراد حذفها فلا تنسخ ببساطة في الملف الجديد. أما السجلات المراد إضافتها فتوضع في الشريط الجديد مدمجه في المواقع الصحيحة لحفظ التتابع الرئيسي. ويتم التجديد وتنسخ السجلات المجددة في ملف الشريط الجديد على التوالي.

عند انتهاء العملية يصبح الملف الرئيسي الجديد أو الشريط الجديد هو الملف الرئيسي أو الشريط الرئيسي. وعادة ما يحفظ الشريط الأصلي كنسخة احتياطية ليستخدم في حالة ما إذا حدث شيء خطأ للشريط الجديد.

الملفات غير التتابعية: هناك مواقف يكون هناك حاجة لعمل ملفات بسيطة خطية غير تتابعية طبقاً لأي حقل. أحد أمثلة استخدام مثل هذا الترتيب هو الركيزة المتتالية زمنياً والتي يحدث فيها أنه في كل مرة يسترجع سجل أو بيانات جديدة فإنها تهبط إلى نهاية الملف. ويقال على البيانات في هذه الحالة أنها مخزنة طبقاً لتسلسل وصولها arrival sequence.

إلا أنه يكون من المفيد كذلك حفظ مثل هذه الملفات غير التتابعية بالنسبة لفئة أوسع من حالات التطبيقات. افتراض إن الملفات غير التتابعية تخدم أحد الأغراض المفيدة في المقام الأول لها الميزة الواضحة التالية. لا يقضي وقت ولا يبذل مجهود لحفظ الملف في ترتيب طبيعي عن طريق الدمج أو الترتيب أو أي معالجة أخرى سواء عندما يتم اعداد البناء لأول مرة أو فيها بعد عندما تضاف سجلات لنهاية الملف. لكن ماذا عن خدمة هذه الملفات لغرض مفيد؟ وإذا لم تكن في تسلسل معين كيف يمكننا أن نجد سجل محدد نبحث عنه؟ من الواضح لجعل ذلك مفيداً بالنسبة للملفات غير التتابعية فيجب أن يكون هناك طريقة لإيجاد سجلات محددة في داخلها بطريقة مباشرة.

هناك طريقة تفكير مخادعة أخرى هي أنه بالرغم من أن مثل هذه الملفات لا تحفظ في أي تسلسل واقعي فيمكن أن تكون مفيدة لكونها قادرة في بعض الأحيان وبالنسبة لبعض التطبيقات على أن تقرأ على التوالي طبقاً لقيم أحد حقولها أو أكثر من حقل من حقولها. وقد يبدو هذا متعارضاً إلا أنه ليس كذلك كما سنرى ذلك قريباً جداً. في واقع الأمر هذه الملفات معتادة جداً في وقتنا الحالي وسوف تناقش

العديد من المرات في هذا الكتاب. آلية توفير الاتصال المباشر وحفظ تتابع هذه الملفات خارجياً مذكورة في القسم التالي:

الملفات المفهرسة:

كما نسبق أن قلنا هناك أسباب قانونية لطلب تخزين سجلات في ملفات بسيطة خطية ليست في ترتيب طبيعي طبقاً لأي حقل بينها في نفس الوقت:

١ - يتوقع أن يكون من الممكن الاتصال بهذه السجلات مباشرة و

٢ - يتوقع أن تسترجع هذه السجلات على التوالي طبقاً لبعض الحقول سواء كان الملف مخزن في ترتيب واقعي طبقاً لهذه الحقول أو لحقول غيرها أم لا.

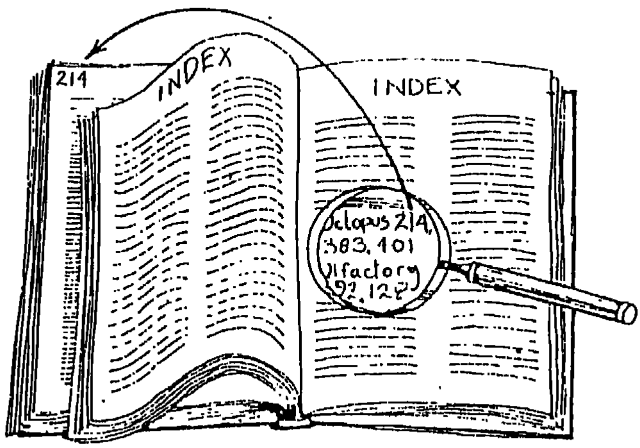
كل هذا يبدو بحيث أننا نكون في حاجة إلى التوسع فيه والقول بأنه من الممكن أيضاً أن يكون هناك رغبة في وجود ملفات تتابعية sequential files يمكن الاتصال بسجلاتها مباشرة ويمكن استرجاع بياناتها مرتبة طبقاً لحقل غير other than الحقل الرئيسي المرتب على أساسه الملف واقعياً. بينما يبدو كل ذلك مثل العداد لما سبق قوله عن الملفات الخطية إلا أنه هناك وحدة يمكن إضافتها للصورة لتحقيق هذه الأهداف بالتمام والكمال. هذه الوحدة تسمى فهرس index.

الشيء المهم بالنسبة لهذه الوحدة هو أنه بينما نوضحها بالطبع، هنا بمصطلحات الكمبيوتر إلا أن الأساس هو نفسه مثل الفهرس الموجود في نهاية هذا الكتاب المكتوب، باللغة الانجليزية، وبعد كل شيء فالكتاب ما هو إلا وسط تخزين لمعلومات عن بعض المواضيع. فكر في أرقام الصفحات وعناوين مجموعات من المعلومات وبالاسم أي كلمات موجودة على هذه الصفحات المحددة. جدول المحتويات الموجود في بداية الكتاب يلخص ما في الكتاب طبقاً للمواضيع الرئيسية ومكتوب بنفس ترتيب مادة الكتاب. لكن ماذا تفعل إذا ما أردت أن تجد جزء معين من المادة أقل من الأجزاء المذكورة في جدول المحتويات دون أن تتصفح صفحات الكتاب كلها؟ ماذا تفعل هو أنك تنتقل الى فهرس الكتاب الموجود في نهايته (كل الكتب المكتوبة باللغة الانجليزية تحتوي على مثل هذا الفهرس: المترجم) وتبحث عن العنصر المطلوب في الفهرس. والفهرس مرتب ترتيباً أبجدياً

طبقاً للعناصر المحتواه في الكتاب وذلك بغض النظر عن مواقع ظهور هذه العناصر في الكتاب.

وكآدميين لدينا المقدرة على عمل البحث في الترتيب الابجدي سريعاً وبكفاءة وذلك في الفهرس لإيجاد الموضوع المطلوب. ماذا بعد ذلك؟ بعد وجود العنصر في الفهرس تظهر أرقام الصفحات.

في واقع الأمر هذه الأرقام ما هي إلا مشير مباشر direct pointer الى صفحة الكتاب التي يظهر فيها العنصر. وتذهب مباشرة إلى هذه الصفحة لتجد المادة التي تبحث عنها هناك شكل 3.3.



شكل 3.3
الفهرس من الكتاب

أبسط أنواع الفهارس لملف البيانات هو الفهرس البسيط الخطي linear index يناقش ويوضح في هذا القسم. وبالطبع يجب أن يبدو هذا المصطلح معتاداً حيث أننا نتكلم عن ملفات بسيطة خطية. وفي واقع الأمر مثل هذا الفهرس ما هو إلا استخدام لمثل هذا النوع من الملفات.

وطبقاً للمناقشة السابقة هناك أربع حالات مميزة لمناقشتها عند استخدام الفهرس (البسيط الخطي أو غيره) مع الملف الخطي البسيط. والأربع حالات مميزة طبقاً لما إذا كان الملف مخزن على التتابع لأحد حقوله أم لا وطبقاً لما إذا كانت

العمليات تشمل استرجاع مباشر لسجل فردي أو استرجاع لكل سجلات الملف مرتبة طبقاً لقيم مجموعة حقول أو أحد الحقول. والرسم والنص التاليان يوضحان الأربع حالات.

	The file is in physical sequence by some field	The file is not in physical sequence by any field
The Concern is direct retrieval of a single record	Case 1	Case 2
Then concern is retrieval of all of the records in order based on the values in some field.	Case 3	Case 4

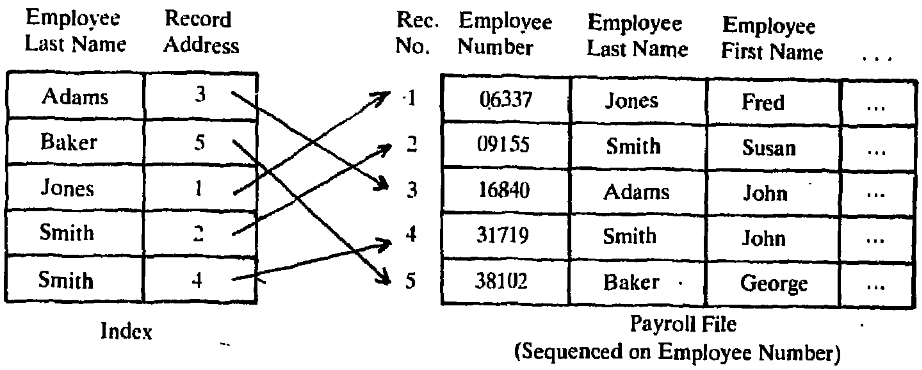
١ ـ الملف مخزن في ترتيب واقعي طبقاً لبعض حقوله وبالإضافة لذلك يلزم اتصال مباشر عن طريق أحد حقوله (والذي يمكن أن يكون نفس الحقل المرتب على أساسه الملف).

عندما يكون حقل البحث search field ـ والذي تستخدم قيمته في تعريف السجل الذي يبحث عنه في الاتصال المباشر ـ هو نفس same حقل التتابع sequence file (الحقل المستخدم في ترتيب الملف واقعياً)، فإن هذه الحالة هي حالة تقليدية في مجال ترتيب الملفات وطرق الاتصال وتسمى ترتيب تتابعي مفهرس indexed - sequential organization. ونظراً لأهميتها التاريخية وتنفيذاتها الممتعة فسوف تعامل على حده في القسم التالي من هذا الفصل. في الحالة الأخرى يخزن الملف في ترتيب واقعي طبقاً لبعض حقوله ويكون الاتصال المباشر مطلوباً إلا أن حقل البحث مختلف different عن حقل التتابع. في هذه الحالة حقيقة أن الملف تتابعي (طبقاً لبغض حقوله) غير مناسبة. فيمكن ألا يكون الملف مخزناً بأي

تتابع واقعي على الإطلاق وعلى هذا فإن الموقف في واقع الأمر هو نفسه مثل الحالة رقم 3 والذي يناقش هنا.

٢ ـ الملف مخزن في تتابع واقعي طبقاً لبعض حقوله وهناك متطلبات بأنه يكون قادر على استرجاع السجلات منه على التتابع طبقاً لبعض حقوله. ومن الواضح إذا كان الحقل الذي سيتم الاسترجاع على أساسه مبنياً على الحقل المرتب على أساسه الملف واقعياً فإن المتطلب يقدر قراءة السجلات طبقاً لتتابعها الواقعي ليس إلا. والحالة الأخرى والأكثر أهمية هي عندما يكون المتطلب هو استرجاع السجلات في تتابع مبني على قيم في حقل غير other الحقل الذي تم إعداد التتابع الواقعي للملف على أساسه. والطرف الأيمن من شكل 3.4 يبين حقول رقم العامل والاسم الأخير للعامل والاسم الأول للعامل الموجودة في ملف الرواتب الموجود في شكل 3.1 (بقية الحقول يفترض وجودها هناك كذلك). لاحظ أيضاً أن سجلات الملف مرقمة 1 و 2 و. . . هذه الأرقام والمعروفة بأنها الأرقام النسبية للسجلات relative record numbers هي صورة واضحة من عنونة السجلات الفردية. وهي الحالة التي تعطي رقم نسبي للسجل وتكون نظم برامج النظام قادرة على عمل بعض حسابات للعناوين وإيجاد السجل المصاحب لها مباشرة. تذكر أن ملف الرواتب هو ملف تتابعي طبقاً لرقم العامل. الطرف الأيسر من شكل 3.4 يبين فهرس خطي بسيط مصمم ليقدم اتصالاً مباشراً لملف الرواتب عن طريق الاسم الأخير للعامل. العمود الموجود على اليسار في الفهرس متطابق مع عمود الاسم الأخير للعامل الموجود في ملف الرواتب فيما عدا أنه مرتب بحيث أن القيم تكون مرتبة ترتيباً أبجدياً. المحتوى الموجود على اليمين في كل صف من صفوف الفهرس يهدف إلى توفير عنوان السجل في ملف الرواتب والذي يحتوي على قيمة حقل الاسم الأخير في صف الفهرس هذا. ويمكن أن يأخذ العنوان شكل عنوان القرص المطلق أو الفعلي أو شكل رقم السجل النسبي كما هو الحال في شكل 3.4. وعلى هذا فإذا رغبنا في استرجاع السجلات من الملف مرتبة أبجدياً طبقاً لآخر اسم للعامل فإننا نقرأ الفهرس سطراً سطراً ببساطة ونستخلص السجل رقم 3 (السجل الموجود فيه Adams) من ملف الرواتب ويتبعه السجل رقم 5 (الذي يحتوي على Baker) وهكذا. لاحظ أنه هناك عاملان لهما نفس الاسم الأخير Smith. هذا هو سبب وجود سطرين في الفهرس لنفس الاسم Smith (دون أي

ترتيب خاص) كل منهما يشير إلى سجل Smith مختلف عن الآخر في ملف الرواتب.



شكل 3.4
ملف تتابعي تحت فهرسته طبقاً لحل آخر

٣ ـ الملف مخزن دون أي تتابع واقعي ويلزم حدوث اتصال مباشر عن طريق أحد حقوله.

يبين شكل 3.5 عينه لسجلات ملف الرواتب الموجود في شكل 3.1 متتابعة واقعياً بحيث أنها ليست في تتابع طبقاً لأي حقل من حقول السجلات. ويبين كذلك فهرساً في الملف الجديد مبنياً على حقل الاسم الأخير للعامل. هذا الفهرس متتابع، ويجب أن يكون كذلك، بالنسبة للاسم الأخير للعامل. افرض أنه هناك حاجة لإيجاد سجل رواتب لمن اسمه Jones مباشرة. أولا يوجد برنامج اسمه برنامج بحث search أو جزء بحث Jones في السطر الثالث من الفهرس. ولا نريد أن نتوقف لمناقشة أجزاء البحث إلا أنها تعمل في مفهومها بنفس طريقة بحث الشخص عن أحد العناصر الموجودة في فهرس أحد الكتب ويصل بسرعة الى العنصر المطلوب. وبعد إيجاد Jones في الفهرس فإن محتوى الفهرس المصاحب لم يحدد أن السجل في الملف الرئيسي الذي له قيمة Jones لحقل الاسم الأخير له رقم نسبي للسجل 2. ويمكن للنظام أن يذهب مباشرة الى السجل رقم 2 في الملف الرئيسي ويستخلص سجل Jones.

Employee Last Name	Record Address
Adams	5
Baker	4
Jones	2
Smith	1
Smith	3

Index

Rec. No.	Employee Number	Employee Last Name	Employee First Name	. . .
1	31719	Smith	John	...
2	06337	Jones	Fred	...
3	09155	Smith	Susan	...
4	38102	Baker	George	...
5	16840	Adams	John	...

Payroll File
(In No Particular Sequence)

شكل 3.5
ملف مفهرس بطريقة غير تتابعية

٤ ـ الملف المخزن بدون أي تتابع إلا أنه هناك متطلب بأنه يكون قادراً على استرجاع سجلاته بتتابع طبقاً لأحد حقوله.

دعنا نعود إلى الملف غير المرتب الموجود في شكل 3.5 وقد بينا كيفية استخدام الفهرس في هذا الشكل لإيجاد سجل في الملف مباشرة. إلا أن نفس هذا الفهرس يمكن استخدامه في استرجاع سجلات الملف بتتابع الحقل الذي بنى على أساسه الفهرس (الاسم الأخير للعامل في هذه الحالة). وذلك بالرغم من أن الملف غير متتابع واقعياً طبقاً لأي حقل من حقوله. وكما في الحالة رقم 2 يتحقق ذلك بالبدء بأول سطر من أسطر الفهرس (والمرتب طبقاً لهذا الحقل) وتحدث القراءة لأسفل سطراً سطراً وفي كل مرة يسترجع السجل من الملف الرئيسي يتحدد ذلك بالعنوان الموجود في سطر الفهرس. في هذه الحالة نسترجع رقم السجل 5 (السجل الخاص بالعامل Adams) أولاً يليه رقم السجل 14 (الخاص بالعامل Baker) وهكذا.

دعنا نراجع بضع نقاط عن الفهارس ونلاحظ بضع ملاحظات في هذا الاتجاه. تكون الخواص افتراضية بطبيعتها عند هذه النقطة ويمكن أن تختلف في التنفيذات الفعلية الموجودة في أيامنا هذه.

- يمكن إعداد فهرس مبنى على أي حقل من حقول الملف سواء كان الملف متتابعاً واقعياً طبقاً لهذا الحقل أو لغيره أو غير متتابع واقعياً على الإطلاق.

- يمكن إعداد فهرس طبقاً لخليط من الحقول. مثـال ذلك في مثال ملف الرواتب يمكن أن نطلب أن نكون قـادرين على إيجـاد سجل مباشرة طبقاً لـلإسم الأخير للعامل وللاسم الأول للعامل. والفهرس في هذه الحالة يشمل كـل من عمودي الاسم الأخير والاسم الأول كما يشمل كذلك العمود اللازم بالعنوان المناظـر في الملف الرئيسي.

- يمكن استخـدام أي فهرس لكـل من الاتصال المبـاشر طبقـاً للحقـل المفهـرس والاسترجاع التتابعي للسجلات طبقاً لتسلسل الحقول أو لتتابعها.

- يمكن أن توجد عدة فهارس منفصلة داخل الملف في نفس الوقت كل منها يعتمد عـلى حقـل مختلف أو عـلى خليط حقـول مختلف من الملف. والفهـارس تكـون مستقلة عن بعضها البعض.

- عنـدما تجـري تغييرات معينـة على الملف يجب أن تجـدد الفهارس. مثـال ذلك إذا تغيرت قيمة أحـد الحقول في أحـد السجلات. فـإذا كان هنـاك فهرس مبني لهـذا الملف على أساس هذا الحقل فيجب عند ذلك تجـديد الفهـرس. مثال ذلـك في شكـل 3.5 إذا تـزوجت Susan Smith وحصلت عـلى الاسم الأخـير لـزوجهـا (Eller) فعلى هذا لا يتجـدد رقم السجل 3 في ملف الـرواتب فقط بل يجب أن يتغير الصف الخامس من الفهرس كذلك. ويجب أن يحل Eller محل Smith في هذا الصف من صفوف الفهـرس ويعاد تـرتيب محتويـات الفهرس أبجـدياً مـرة أخرى. ومن الواضح أن قيم الحقل في الفهـرس يجب أن تتواءم طـوال الوقت مع قيم الحقل المناظر في الملف الرئيسي لكي يكون لمفهوم الفهرس معنى. وهذا صحيـح أيضـاً إذا أدخـل سجـل جـديـد داخـل الملف أو إذا حـذف أحـد سجلات الملف الموجودة به منه. فيجب أن يعاد ترتيب الفهرس فعلياً بالبيـانات الجديدة أمـا فور حـدوثه أو بعـد فترة زمنية معينة. إذا حـدث ذلك بعـد انقضاء فترة زمنية معينة والتي كما سوف ترى تكون ميزة في العادة بالنسبـة للأداء فيجب أن يحفظ التجديد في قائمة يمكن الاتصـال بها بـطريقة مـا في حالـة ما إذا كـان يجب عمـل استرجـاع مبنى على أحـد المحتويـات الجديـدة قبل أن يعـاد تـرتيب الفهرس.

الملفات التتابعية المفهرسة

مقدمة: كما سبق أن ذكرنا في آخر قسم عندما يحدث تتابع للملف طبقاً لحقل معين ويعد فهرس لهذا الملف مبني على نفس same الحقل فيعرف الترتيب بأنه ملف تتابعي مفهرس indexed - sequential file. ومميزات هذا التكوين واضحة لتطبيقات معينة. مثال ذلك، اعتبر ملف الرواتب المعد بصورة تتابعية مفهرسة مع استخدام رقم العامل كحقل تتابع وكحقل يبني على أساسه الفهرس. باستخدام نفس هذا الملف الوحيد يمكننا انتاج شيكات رواتب على التتابع طبقاً لترتيب رقم العامل وذلك كل يوم جمعة، وباستخدام الفهرس يمكننا أن نوجد سجلات فردية مباشرة لتجديدها (مثال ذلك عند زيادة راتب العامل 31719).

الملفات التتابعية المفهرسة لها تاريخ طويل في تشغيل البيانات وقد نشأت كتوسع طبيعي للملفات التتابعية البسيطة. وقد درس وجهان من أوجه الملفات التتابعية المفهرسة بعمق على مدار السنين ووجد عدد من البدائل لهذه الملفات. هذين الوجهين هما تكوين الفهرس وطريقة إضافة السجلات الجديدة للملف بحيث يمكن حفظ كل من الطبيعة التتابعية والاتصال المباشر معاً. سوف نلقي نظرة على ثلاثة طرق لتكوين ملفات تتابعية مفهرسة مع تركيز الانتباه الخاص لتكوين الفهرس ولطريقة إدخال السجلات الجديدة. والثلاث طرق هي الفهرس البسيط والفهرس الموجهة لنظم المكونات متعددة المستوى والفهرس الموجهة لشجرة B.

الفهرس البسيط

لقد بين شكل 3.4 ملف الرواتب في تتابع طبقاً لحقل رقم العامل مع فهرس مبني على حقل مختلف من حقول هذا الملف. يبين شكل 3.6 نفس الملف والذي ما زال في تتابع طبقاً لحقل رقم العامل ولكن مع فهرس خطي بسيط مبني على نفس same الحقل. مثل هذا الترتيب المستخدم لفهرس خطي بسيط ربما يكون أبسط أنواع الملفات التتابعية المفهرسة. ويجب أن يكون واضحاً عند هذه النقطة أن السجلات الموجودة في الملف الموجود في شكل 3.4 يمكن أن يتم الاتصال بها تتابعياً بواسطة رقم العامل حيث أنها مخزنة واحداً تلو الآخر في مجموعة أقراص بنفس هذا الترتيب. كما يمكن استرجاعها مباشرة أيضاً بواسطة

رقم العامل باستخدام الفهرس وذلك باستخدام جزء بحث أو برنامج بحث مناسب.

Employee Number	Record Address
06337	1
09155	2
16840	3
31719	4
38102	5

Index

Rec. No.	Employee Number	Employee Last Name	Employee First Name	...
1	06337	Jones	Fred	...
2	09155	Smith	Susan	...
3	16840	Adams	John	...
4	31719	Smith	John	...
5	38102	Baker	George	...

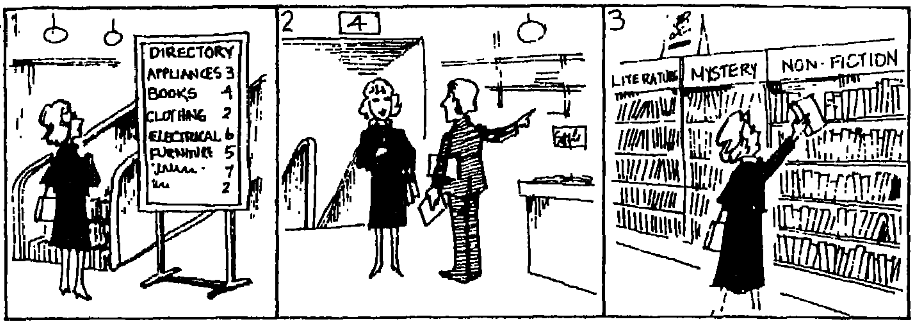
Payroll File
(Sequenced on Employee Number)

شكل 3.6
ملف تتابعي أعدت فهرسة له طبقاً لحقل التتابع

قد تتعجب ما الفائدة من انتاج فهرس منفصل يعيد أحد الحقول من الملف الأصلي بنفس الترتيب. إذا استطعت أن تطبق برنامج أو جزء البحث على مثل هذا الفهرس لإيجاد سجل على أساس مباشر لماذا لا تطبق نفس الجزء أو البرنامج للبحث على الملف الرئيسي وتوفر كل هذا المجهود في عمل فهرس منفصل؟ تقع النتيجة في الحقيقة بأن أي شيء يبحث عنه يسحب داخل الذاكرة الرئيسية لعمل البحث الفعلي، وكلما ازداد حجم الملف كلما أصبح الفهرس كبيراً وكلما أصبح الفهرس كبيراً كلما أصبح عمل البحث خلال الفهرس لإحدى القيم المطلوبة كبيراً، فإذا كان كبيراً جداً بحيث لا يمكن وجوده داخل الذاكرة الابتدائية في نفس الوقت فيجب إعداد مخطط لإدخال اجزاء منه داخل الذاكرة الابتدائية وإخراجه منه وذلك ومن وإلى الذاكرة الثانوية أثناء عملية البحث ويمكن أن يصبح ذلك عنق زجاجة سريعاً وغير متوقعاً. وإذا ما أريد لمحتويات السجلات غير حقول البحث أن توجد في الذاكرة الابتدائية للبحث فإن المشكلة تتعقد الى درجة أسوأ. (بالطبع بالنسبة للملف الكبير يراد إحضار عدد كبير من سجلات الفهرس داخل الذاكرة الابتدائية بحيث أنه لا يعني شيء صغر الحجم وهذا الترتيب يكون غير عملي).

وبالنسبة لسؤال إدخال سجلات جديدة داخل الملف فإن الموقف هو نفسه مثل حالة الملفات التتابعية التي سبق مناقشتها. فيجب إعادة كتابة الملف مع إدخال السجلات الجديدة في تسلسلها الصحيح أو أنه يعاد ترتيبها في مواقعها. ويجب أن توضع محتويات السجلات الجديدة بالطبع في التتابع الصحيح في الفهرس مع ضرورة اجراء عملية ترتيب هناك في الفهرس كذلك.

فهرس موجه لنظم مكونات متعددة المستوى: لاعتبار نوع آخر من أنواع الفهارس اعتبر الموقف الحالي. دعنا نقول أنك تذهب إلى متجر كبير مكون من ثمانية طوابق لشراء نسخة من كتاب معين شكل 3.7. عليك بالمرور داخل كل



شكل 3.7
مستويات البحث في أحد متاجر بيع الكتب

الممرات في كل دور بحثاً عن الكتاب وحتى المشتري الأكثر التزاماً يجد في ذلك ارهاق غير لازم. الشيء المحتمل أكثر هو أن أول شيء تفعله عند دخولك المتجر هو البحث في دليل المتجر. وهذا الدليل يسرد في ترتيب أبجدي كل الأقسام المختلفة للمتجر ويحدد موقع كل منها. وسوف تجد مدخل الكتب وتلاحظ أنه موجود في الطابق الخامس. وعلى هذا فإنك تصعد إلى الطابق الخامس. وهناك يوجد عدد من الأقسام المختلفة. وبدلاً من الحركة داخل ممرات هذا الطابق بحثاً عن قسم الكتب فإنك تسأل أحد العاملين في المتجر عن موقع هذا القسم. ويشير لك إلى القسم المطلوب لتذهب إليه. ويمكن أن تبحث في كل رف من أرفف قسم الكتب حتى تجد الكتاب المطلوب لكنك تجد أسماء موضوعة للمواضيع مثل كتب الطبخ والخيال العلمي وما إلى ذلك. تتبع العلامات لتحدد الاسم الذي يشمل

الموضوع الذي تبحث عنه. والكتب تحت كل اسم مرتبة ترتيباً أبجدياً طبقاً لاسم المؤلف وعليك بتتبع الأرفف التي تقع تحت هذا الاسم حتى تجد الكتاب المطلوب.

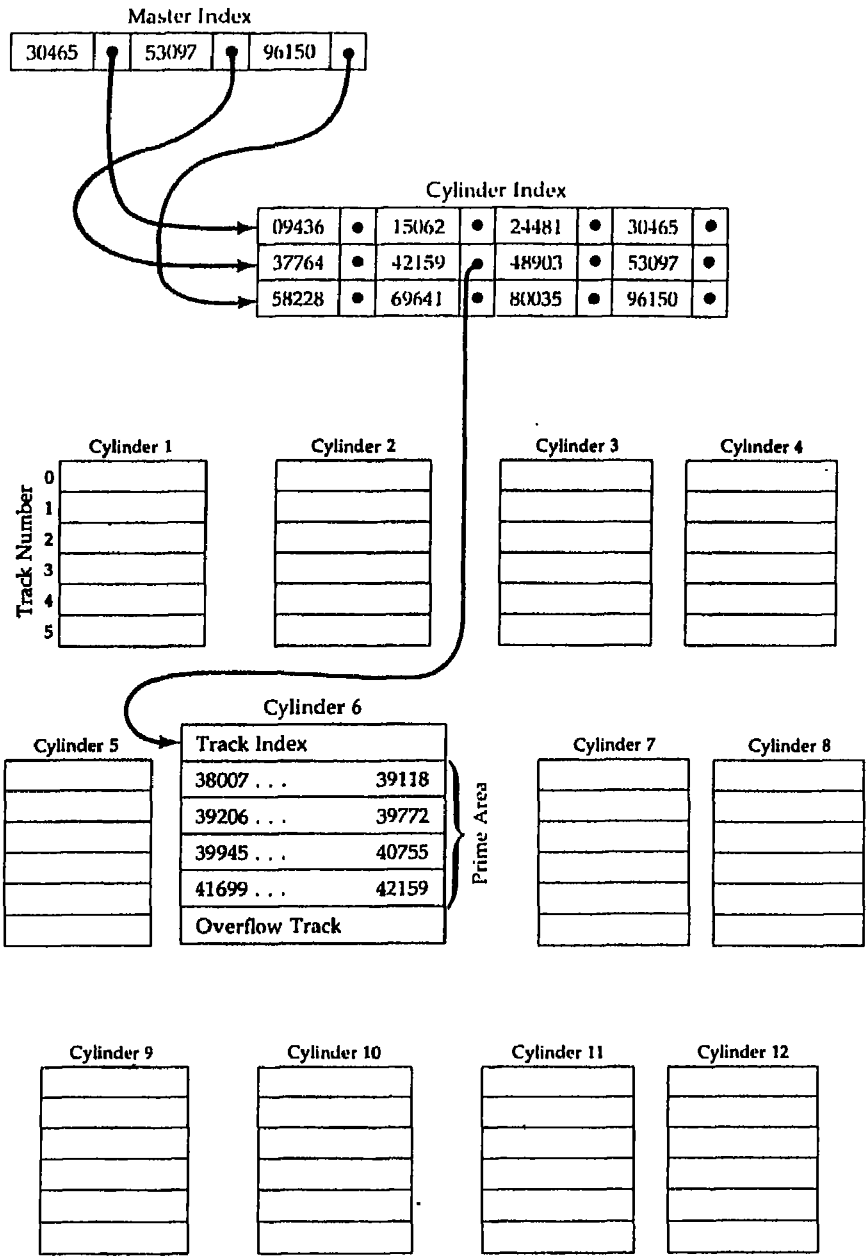
فكر في سلسلة الأحداث. إنك قمت بتضييق البحث على التتابع طبقاً لخواص طبيعية للبيئة (أولاً تحديد الطابق وبعد ذلك الأقسام الموجودة في الطابق ثم تحديد الموضوع) حتى وصلت إلى مستوى صغير بدرجة كافية يجعلك تصل بسهولة الى الشيء المطلوب. وهذا بكل تأكيد أوفر كثيراً من البحث في كل عنصر من العناصر في كل طابق من طوابق المتجر بدءاً بأول طابق وصاعداً لأعلى.

مثال لاجراء بحث متماثل مع سجل في ملف كمبيوتر هو طريقة الاتصال التتابعي المفهرس لشركة Indexed Sequential Access Method (ISAM) IBM فهي تقرب كذلك أو تضيق على التتابع من البحث عن العنصر المطلوب (السجل) وذلك طبقاً لخواص طبيعية للبيئة (مجموعة الأقراص).

ويبين شكل 3.8 ملف أفراد ومعه فهرس من ثلاثة مستويات مصاحب له ومرتب طبقاً لطريقة الاتصال التتابعي المفهرس ISAM (ثالث مستوى للفهرس مخزن في خليط من المسار رقم 0 لكل الاسطوانات). ويحتوي كل سجل على عدد من الحقول: رقم العامل المكون من خمسة أرقام والذي يعمل كحقل رئيسي والاسم الأخير للعامل وما إلى ذلك. ويحدد أرقام للعاملين بالشركة بحيث أن أسماؤهم لا تكون في تتابع مع مجموعة أقراص بها ستة أسطح تسجيل وهذا يعني وجود ستة مسارات في الاسطوانة (مرقمة من 0 إلى 5).

تكون السجلات مخزنة تتابعياً في البداية وذلك طبقاً لحقل رئيسي في المنطقة الأولى prime area وهي المسارات من 1 إلى 4 في شكل 3.8 من كل من الاثنى عشر اسطوانة. يبين شكل 3.9 الاسطوانة 6 بتفاصيل أكثر. يوجد 16 سجل في ملف الأفراد هذا لها قيم للحقل الرئيسي تتراوح من 38007 وإلى 42159 وتقع في المنطقة الأولى للإسطوانة 6 عندما تم تحميل الملف. وقبل الدخول في موضوع الفهرس لاحظ وتذكر أنه يمكن قول أي شيء عن ترتيب ISAM إلا أن البيانات موضوعة تتابعياً في المنطقة الأولى وبقراءة مسار كل اسطوانة من المنطقة الأولى على

التتابع يمكن استرجاع الملف بسهولة سجلاً سجلاً ومرتباً طبقاً للحقل الرئيسي رقم العامل.



شكل 3.8

ملف ISAM من شركة IBM الفهرس والبيانات

Normal Entry for Track 1 — Overflow Entry for Track 1

Track Index	40755 T3 40755 XX 39772 T2 39772 XX 39118 T1 39118 XX 42159 T4 42159 XX			
	38007 Jones ...	38642 Adams ...	38750 Taylor ...	39118 Hall ...
Prime	39206 Carter ...	39353 Newton ...	39541 Jones ...	39772 Allen ...
Area	39945 Smith ...	40230 James ...	40617 Sloan ...	40755 Potter ...
	41699 Baker ...	41837 Dodd ...	41922 Logan ...	42159 Smith ...
Overflow Track				

شكل 3.9
ملف ISAM من شركة IBM الاسطوانة 6 بالتفصيل

دعنا نرى الأن الفهرس متعدد المستويات (والمسمى كذلك بغير الخطى) والذي يقدم للاسترجاع المباشر للسجلات طبقاً لحقل رقم العامل. وكما يوضح شكل 3.8 فهناك ثلاثة مستويات للفهرس في الأساس يحتوي كل منها على فئة جزئية من مفاتيح الملف. ويحفظ كل مستوى فهرس مفاتيحه على التوالي. ويسمى المستوى الأول الفهرس الرئيسي master index ويمكن أن يفترض أنه في الذاكرة الابتدائية. وتقود المشيرات من الفهرس الرئيسي الى مجموعة من مشيرات الاسطوانة في فهرس الاسطوانة cylinder index. وفي شكل 3.8 يوجد لفهرس الاسطوانة ثلاثة من هذه المجموعات (كل صف هو مجموعة في حد ذاته) وكل مشير في فهرس الاسطوانة يشير بصورة فردية الى اسطوانة خاصة. ويسمى أول مسار (المسار رقم 0) من كل اسطوانة بفهرس المسار track index ويقود إلى مسار معين على هذه الاسطوانة حيث يوجد السجل الذي له قيمة معينة لمفتاحه.

دعنا نأخذ مثالاً ونرى كيف تعمل هذه العملية مع الملف المبين في شكل 3.8. افرض أن السجل المطلوب خاص بالعامل رقم 40230. تتبع طريقة الاتصال الفهرس الرئيسي من اليسار الى اليمين بحثاً عن أول قيمة للمفتاح أكبر higher من 40230 وهي 53097 وبعد ذلك يتبع المشير المصاحب للقيمة 53097 إلى الصف الثاني أو إلى مجموعة قيم في فهرس الاسطوانة. وتتكرر العملية مع تتبع مجموعة من المفاتيح من اليسار الى اليمين بحثاً عن أول قيمة أكبر من 40230 وهي

42159. وعند ذلك فإنه يتبع المشير المصاحب لهذه القيمة إلى بداية الاسطوانة 6 كما هو مبين في الشكل. أول مسار لهذه الاسطوانة (المسار 0) يحتوي على فهرس مسار له. وبالنقل إلى شكل 3.9 فإننا نرى أن فهرس المسار يحتوي على محتوى معتاد normal entry ومحتوى سريان زائد overflow entry لكل مسار في المنطقة الأولى من الاسطوانة. ويحتوي المحتوى المعتاد على أعلى قيمة للمفتاح على هذا المسار ورقم المسار (أو العنوان) كما يحتوي محتوى السريان الزائد أساساً على أعلى قيمة للمفتاح على هذا المسار ومحتوى وهمي (مبين على أنه XX في شكل 3.9). ويتتبع النظام قيم المحتوى المعتاد بحثاً مرة أخرى عن أول قيمة أعلى من 40230. وهذه هي 40755. ويحدد المشير المصاحب للقيمة 40755 إن السجل الذي له قيمة مفتاح 40230 يوجد في المسار 3 من هذه الاسطوانة. وبعد ذلك يتتبع النظام المسار 3 حيث يجد السجل الذي له قيمة مفتاح 40230. وكما في حالة مثال المتجر قد يبدو ذلك معقداً في البداية إلا أنه أكثر كفاءة بالمقارنة بإجراء البحث المتتابع عن نفس السجل.

وكما تنظر إلى شكل 3.9 فقد تحدث لك مشكلة. فهو يشبه البيانات كما سبق تحميلها في البداية حيث يملأ كل مسار من مسارات المنطقة الأولى وهذا صحيح. عند إضافة سجلات جديدة إلى الملف فأين تذهب هذه السجلات وكيف يظل الملف متتابعاً وكيف يحدث الاتصال المباشر لكل السجلات المحفوظة؟ سؤال جيد.

تبدأ الإجابة بمسارات السريان الزائد الموجودة في نهاية كل اسطوانة (وكذلك مسارات أخرى تسمى بمناطق مستقلة للسريان الزائد). وتشمل محتويات السريان الزائد في كل فهرس مسار. افرض أننا نريد إضافة سجل جديد له قيمة مفتاح 38693 للملف. سوف يستمر النظام خلال الثلاث مستويات للفهرس كما لو كان يحاول أن يجد سجلاً له قيمة المفتاح. وباتباع نفس طريقة البحث التي سبق ذكرها فإن النظام يقرر إذا كان يضع السجل في المسار 1 من الاسطوانة 6، انظر شكل 3.9، أم لا. لكنه يجد بالطبع أن هذا المسار مملوء. وعلى هذا يستمر النظام في إدخال السجل الجديد في موقعه الصحيح على المسار مع ترحيل السجلات الأخرى ناحية اليمين وإدخال آخر سجل من سجلات المسار في منطقة

السريـان الزائـد كما هـو مبين في شكـل 3.10 وتتغير قيم فهـرس المسار للمســار 1 بحيث أن:

١ ـ المحتوى المعتاد يبين أعلى قيمة مفتاح جديدة لكـل السجلات المـوجودة عـلى المسار (وهي 38750 الآن).

٢ ـ محتوى السريان الزائد يبين:

أ ـ آخر last سجل سريان زائد لهذا المسار طبقـاً لقيمة المفتـاح key value (وهي 39118) و.

ب ـ أول first سجل سريان زائد لهذا المسار طبقاً للعنـوان address (وهي المسار 5 والسجل 1).

38750 T1 39118 T5 Rec 1	39772 T2 39772 XX	40755 T3 40755 XX	42159 T4 42159 XX
38007 Jones ...	38642 Adams ...	38693 Hughes ...	38750 Taylor ...
39206 Carter ...	39353 Newton ...	39541 Jones ...	39772 Allen ...
39945 Smith ...	40230 James ...	40617 Sloan ...	40755 Potter ...
41699 Baker ...	41837 Dodd ...	41922 Logan ...	42159 Smith ...
39118 Hall ... T1			

شكل 3.10
ملف ISAM من شركة IBM الاسطوانة 6 مع إضافة السجل 38693

يستمـر شكل 3.11 في التـوضيح بـإضافـة السجـل الّـذي لـه قيمـة مفتـاح 38444. ويوضع على نفس المسار 1 مع دفع السجل الذي لـه قيمة مفتـاح 38750 في السريـان الزائـد للمسار. وبحفظ التسلسـل يظهر فهـرس المســار 38693 الآن كـأعلى قيمـة مفتاح عـلى المسار 1 المنـاسب و 39118 كقيمة مفتـاح لآخـر سجـل مـوضوع في السريـان الزائـد للمسار، والمسار 5 والسجـل 2 كعنـوان لأول سجـل سريان زائد لهذا المسار. لاحظ كذلك أن كل سجلات السريان الزائد لمسار معين هي متسلسلة chained مع بعضها في قائمة متصلة. وكما هو مبـين في شكل 3.11 إذا لم يكن السجل اللازم وجوده على المسار 1 من هذه الاسـطوانة مـوجوداً هنـاك

(وله قيمة أعلى من 38693) فيشير فهرس المسار الى السجل الذي لـه قيمة مفتـاح 38750 أولاً. فإذا لم يكن هذا هو السجل المطلوب فإنه لديـه مشير مصـاحب معه يشـير إلى السجل التـالي على التـوالي والذي يجب أن يكـون على المسـار 1 إذا كان هناك مكان كاف لذلك وهكذا. ومنطق محتويات السريان الزائد في فهـرس المسار يبدو أكثر تعقيداً عما هو الحال فعلاً. ففي الواقع أنه يقول لك أين where يـوجد أول سجل سريان زائد للمسار وما هو آخر سجل سريان زائد للمسار.

38693 TI 39118 T5 Rec 2	39772 T2 39772 XX	40755 T3 40755 XX	42159 T4 42159 XX
38007 Jones ...	38444 Evans ...	38642 Adams ...	38693 Hughes ...
...	39353 Newton ...	39541 Jones ...	39772 Allen ...
39945 Smith ...	40230 James ...	40617 Sloan ...	40755 Potter ...
1699 Baker ...	41837 Dodd ...	41922 Logan ...	42159 Smith ...
39118 Hall ... T1	38750 ... T 5 Rec 1		

شكل 3.11

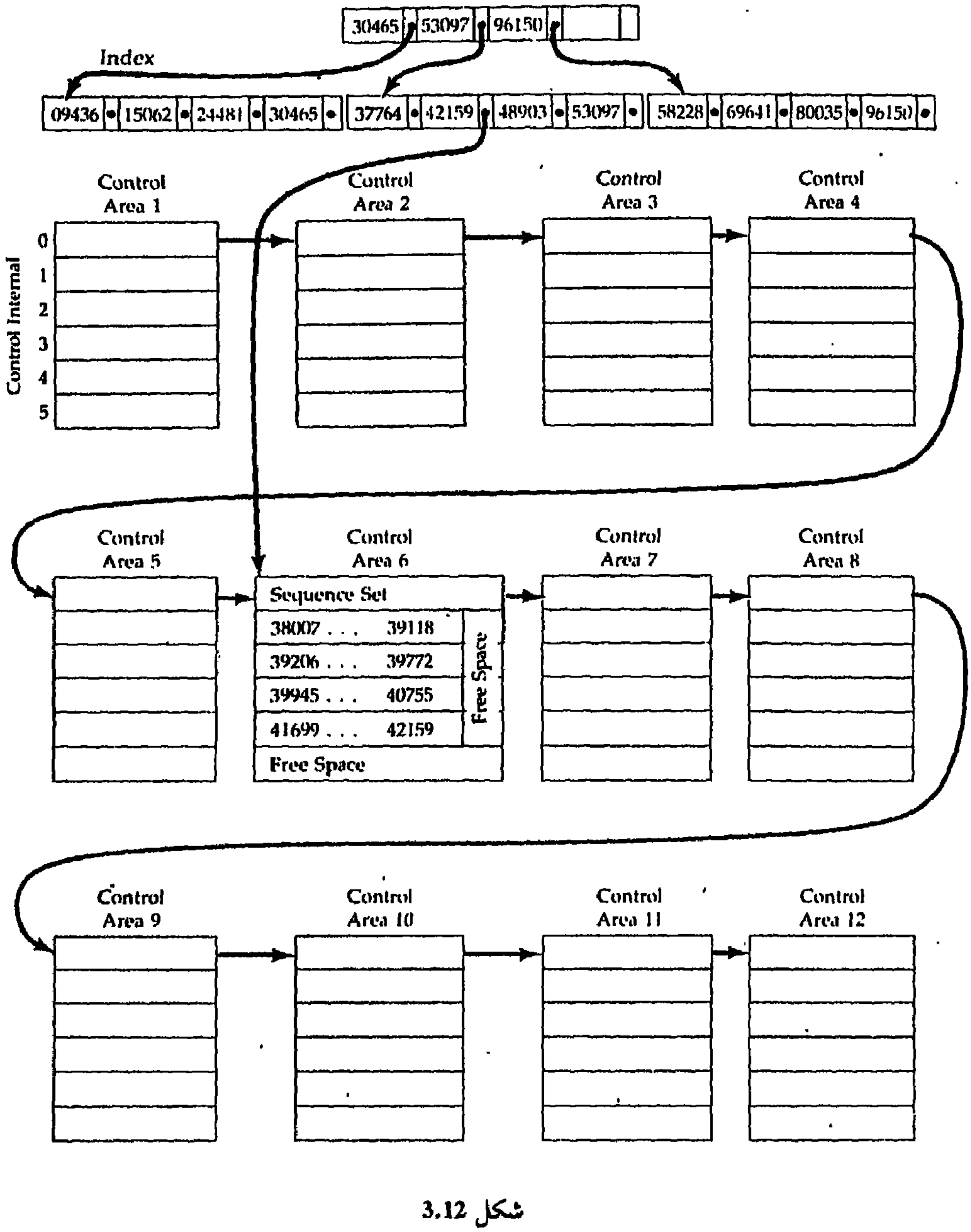
ملف ISAM من شركة IBM: الاسطوانة 6 مع إضافة السجل 38444

يجب أن يستخـدم النظام تـرتيب السريان الـزائد بـوضوح وذلـك في عمـل استرجاعات تتابعية متتالية وكذلـك في عمل بحث مباشر. ومع إضـافة سجـلات أكـثر للملف تزداد مسـارات المنطقـة الأولى التي يكون لهـا سجـل واحـد أو أكـثر مرحلين في سريانها الزائد. ويستخـدم السريان الـزائد للمسـار الموجـود على نفس الاسطوانة أولاً وعندما يمتـلأ تستخدم مسـارات السريان الـزائد المستقلة المـوجودة على اسطوانات أخرى. وسوف تصبح عمليات الاسترجاع والبحث أبطأ حيث أنه يكون هناك حاجة لقراءة مسارين في حـالة مـا إذا استخدم مسـار السريان الـزائد المعتاد وتقفز آلية ذراع الاتصال عـبر اسطوانـات مختلفة في حـالة مـا إذا كان يجب استخدام مسارات سريان زائد مستقلة. وللمساعدة في حـل هذه المشكلة يجب أن يعاد ترتيب الملف بين الحين والآخر. وفي هذه العملية تقرأ محتـويات الملف سـواء كـانت سجلات السريـان الزائـد أو السجلات الأخـرى في ترتيب رئيسي مـع بناء مجموعة جديدة من الفهارس المناسبة.

الفهرس الموجه لشجرة B: بينما كان مفهوم ISAM للفهارس متعددة المستوى خطوة مهمة للأمام في تاريخ تشغيل البيانات إلا أنه له بعض أوجه القصور. لأحد الأشياء فهو يعتمد داخلياً على الخواص الطبيعية لوحدات التخزين والتي تحد من مرونته. وطريقة معالجة سجلات السريان الزائد تجعل الاسترجاع مرهقاً مع تزايد عدد سجلات السريان الزائد. وفي نفس الوقت فإنه يتطلب إعادة ترتيب متكررة لنقل سجلات السريان الزائدة إلى المساحات الأولى.

وقد طور نوع أحدث آخر من أنواع ملفات التتابع المفهرس مستقل عن نظم البرامج وقادر على اكتسابه سجلات جديدة بطريقة أقل ارهاقاً عن طريقة ISAM. وهو مبني على تكوين معروف بأنه شجرة B (B - tree) ومثال هذا الترتيب الحديث هو فئة البيانات الرئيسية المتتابعة Key Sequenced Data Set (KSDS) لطريقة اتصال التخزين الافتراضي Virtual Storage Access Method (VSAM) لشركة IBM. اصطلاح طريقة اتصال التخزين الافتراضي VSAM في واقع الأمر خطأ في التسمية حيث أنه تجميع فعلي لثلاثة طرق تخزين. إحدى هذه الطرق هي طريقة KSDS والطريقتان الأخريتان هما فئة البيانات المحتواه التتابعية Entry Sequenced Data Set (ESDS) والمصممة للملفات المتتابعة وفئة بيانات السجلات النسبية Relative Record Data Set (RRDS) والمستخدمة في ملفات التكرار (انظر القسم التالي). يبين شكل 3.12 ترتيب KSDS وينقسم مكان التخزين في KSDS الى خطوات مراقبة control intervals ومناطق مراقبة con- trol areas. وخطوة المراقبة هي وحدة نقل للبيانات بين الذاكرة الابتدائية والذاكرة الثانوية. ويمكن أن تكون، وليس هذا شرطاً، مسار قرص. وتتكون منطقة المراقبة من عدد من خطوات المراقبة. ويمكن أن تكون، وليس هذا شرط، اسطوانة قرص. والفهرس متعدد المستويات مثل فهرس ISAM يكون هرمي الشكل بطبيعته ويمكن أن يزداد عمقه مع نمو الملف. وهناك سجل فهرس واحد في أعلى مستوى ويخرج السجل لأسفل من عند هذه النقطة العلوية. ويعتمد عدد مستويات الفهرس على حجم الملف وعلى حجم سجلات الفهرس. وكل سجلات الفهرس ذات المستوى الأدنى من الفهرس تعرف مع بعضها على أنها فئة تتابع sequence set. وهناك خيار لكل سجل فئة تتابع لتخزينه مع منطقة

التحكم التي يفهرسها وهذا لأسباب أداء كما هو مبين في شكل 3.12.



شكل 3.12

ملف KSDS من شركة IBM: الفهرس والبيانات

وتعد محتويات الفهرس عند مستوى معين للفهرس لتمثل أعلى محتوى

فهرس لسجلات الفهرس الموجودة عند المستوى التالي للفهرس مع الإشارة لسجل الفهرس. مثال ذلك في شكل 3.12 يشير المحتوى 30465 الموجود في سجل فهرس أعلى مستوى إلى سجل فهرس في المستوى التالي لهذا المستوى والذي ينتهي بنسخة خاصة منه 30465. ويشير كل محتوى لسجل فئة التتابع الى خطوة مراقبة فردية لمنطقة مراقبة ملف البيانات التي يفهرسها وبالتماسك مع مخطط الفهرسة تحتوي أعلى قيمة مفتاح لسجلات البيانات في خطوة المراقبة هذه. وهذا موضح لمنطقة التحكم 6 في شكل 3.13.

Entry for
Track 1

Sequence Set	39118 T1	39772 T2	40755 T3	42159 T4	free T5
	38007 Jones ...	38642 Adams	38750 Taylor	39118 Hall	free space
	39206 Carter	39353 Newton	39541 Jones	39772 Allen	free space
	39945 Smith	40230 James	40617 Sloan	40755 Potter	free space
	41699 Baker	41837 Dodd	41922 Logan	42159 Smith	free space
	free control interval				

شكل 3.13
ملف KSDS من شركة IBM: منطقة التحكم 6 بالتفصيل

وسوف نعود كما سنرى فيما بعد إلى أنه بينما تكون البيانات الموجودة في خطوة تحكم معينة متتابعة دائماً إلا أن خطوات التحكم مع مرور الوقت لا تصبح متتابعة وذلك بسبب إدخال بيانات جديدة ويصبح السؤال الآن: كيف يمكن استرجاع الملف في تتابع رئيسي؟ والإجابة هي أن سجلات فئة التتابع تحفظ دائماً التتابع الصحيح وتكون جميع سجلات فئة التتابع متصلة مع بعضها بواسطة فئة من المشيرات الأفقية horizontal pointers كما هو مبين في شكل 3.12 وبتتبع سلسلة سجلات فئة التتابع مع اهمال مستويات الفهرس العلوية يمكن استرجاع الملف في تتابع رئيسي.

مرة أخرى كما في ISAM لإيجاد سجل على أساس مباشر يبدأ النظام على يسار سجل فهرس فردي في قيمة هرمية الفهرس ويفحص متجهاً لليمين مع

البحث عن أول قيمة أعلى من قيمة مفتاح السجل المراد البحث عنه. وبعد ذلك يتبع المشير لأسفل إلى سجل فهرس عند المستوى التالي للفهرس ويستمر في هذه العملية حتى يصل إلى خطوة تحكم ملف البيانات الصحيحة التي يفحصها بهدف الحصول على السجل المراد.

والجمال الفعلي في KSDS يقع في طريقة إدخال السجلات الجديدة في الملف. ففي ISAM يتسبب الإدخال في زيادة وجود ممرات اتصال طويلة لبعض السجلات مع نمو سلاسل السريان الزائد أما في KSDS فإن كل السجلات يكون لها نفس طول مسارات الاتصال بغض النظر عن عدد السجلات الجديدة التي يتم إدخالها. كما أن إعادة الترتيب المتكررة المطلوبة في ملفات ISAM تقل. واحد الأشياء التي يجعل ذلك ممكناً هو الطريقة التي تترك بها الأماكن الفارغة في الملف عند تحميلها. يبين شكل 3.13 مكاناً فارغاً متروكاً في كل خطوة تحكم تحتوي على سجلات بيانات وخطوة تحكم واحدة متروكة فارغة. كما يوجد كذلك مناطق مراقبة فارغة. هذه الأماكن الفارغة، والتي تستخدم كما هو موصوف فيما بعد، تكون أساساً لأسلوب إدخال السجلات في نظام KSDS.

39118 T1	39772 T2	40755 T3	42159 T4	free T5
38007 Jones ...	38642 Adams	38693 Hughes	38758 Taylor	39118 Hall ...
39206 Carter	39353 Newton	39541 Jones	39772 Allen	free space
39945 Smith	40230 James	40617 Sloan	40755 Potter	free space
41699 Baker	41837 Dodd	41922 Logan	42159 Smith	free space
free control interval				

شكل 3.14

ملف KSDS من شركة IBM: منطقة التحكم 6 مع إضافة السجل 38693

ويبين شكل 3.14.منطقة التحكم 6 مع إضافة السجل الذي له قيمة مفتاح 38693. وطبقاً لتتابع المفتاح يجب أن يوضع هذا السجل في خطوة التحكم أو في واقع الأمر يمكن أن يحدث ذلك لأنه هناك مكان فارغ كاف في خطوة التحكم لعمل ذلك. ويدخل السجل الجديد بين السجلات التي لها قيم مفتاح 38642

و 38750 وترحل السجلات التي لها قيم مفتاح 38750 و 39118 الى اليمين بحيث أن إدخال السجل لا يؤثر على تتابع الملف.

يبين شكل 3.15 منطقة التحكم 6 مع إضافة سجل له قيمة مفتاح 38444. وكما في الحالة الأخيرة يجب أن يدخل السجل في خطوة المراقبة 1. إلا أنه لا يوجد مكان في هذه الحالة له فقد سبق استخدام كل الأماكن الفارغة الموجودة في خطوة التحكم بالفعل. وعلى هذا يقوم النظام بعمل ما نسميه بالفصل splitting. ويبدأ ذلك بالنظر الى خطوة فارغة تماماً تكون موجودة في منطقة التحكم. فإذا وجد النظام إحدى هذه الخطوات فإنه يقسم كل البيانات على خطوة التحكم كاملة بالإضافة إلى السجل الجديد مع ترك حوالي نصف البيانات في خطوة التحكم الأصلية ووضع النصف الآخر في خطوة التحكم التي وجدها فارغة.

وعلى هذا ففي شكل 3.15 تظل السجلات التي لها مفاتيح 38007 و 38444 (السجل الجديد) و 38642 في خطوة التحكم بينما السجلات التي لها مفاتيح 38693 و 39118 38750 تنقل كلها إلى خطوة التحكم 5. وتتغير فئة التتابع كما هو مبين في هذا الشكل بحيث أن المفاتيح الموجودة بها والتي تمثل أعلى مفتاح في كل خطوة مراقبة تظل متتابعة حتى بالرغم من أن خطوات التحكم لا تظل متتابعة.

38642 T1	39118 T5	39772 T2	40755 T3	42159 T4
38007 Jones	38444 Evans	38642 Adams	free space	
39206 Carter	39353 Newton	39541 Jones	39772 Allen ...	free space
39945 Smith	40230 James	40617 Sloan	40755 Potter ...	free space
41699 Baker	41837 Dodd	41922 Logan	42159 Smith ...	free space
38693 Hughes	38750 Taylor	39118 Hall	free space	

شكل 3.15

منطقة التحكم 6 مع إضافة السجل 38444 من ملف KSDS من شركة IBM

ماذا يحدث إذا استمرت هذه العملية حتى النقطة التي يراد إدخال سجل جديد فيها في خطوة تحكم كاملة ولا توجد خطوات تحكم فارغة في منطقة التحكم هذه يمكن إجراء تجزئة عليها؟ عند ذلك يتم الحصول على منطقة تحكم كاملة

جديدة وتقسم محتويات منطقة التحكم الجديدة. ويظل حوالي نصف بيانات منطقة التحكم عليها وينقل النصف الآخر الى منطقة التحكم الجديدة. ولكن كما نتذكر من شكل 3.12 يجب أن يشار إلى كل منطقة تحكم من المنطقة التالية الى المستوى الأدنى للفهرس (المستوى الموجود بعد فئة التتابع مباشرة). وعلى هذا فإضافة منطقة تحكم جديدة الى الملف تعني أنه يجب أن يضاف محتوى جديد إلى هذا المستوى من مستويات الفهرس. هذا حسن طالما أنه هناك مكان في سجل الفهرس المناسب. فإذا لم يكن هذا هو الحال فإنه ينقسم متسبباً في إضافة محتوى آخر لمستوى الفهرس أعلاه. ويمكن أن يحتوي أعلى مستوى فهرس فقط على سجل فهرس واحد. وعلى هذا فإذا حدثت هذه العملية للخلف حتى أعلى مستوى فهرس ووجدت أن سجل الفهرس الموجود هناك مملوء فيجب أن يقسم النظام سجل الفهرس هذا ويضيف سجل فهرس أخر عند مستوى أعلى جديد (والذي يصبح المستوى الأعلى أو مستوى القمة الجديد) للفهرس.

ملفات التكرار

هناك العديد من التطبيقات التي تحدث فيها كل الاتصالات بالملفات على أساس مباشر وتكون السرعة مهمة. بالنسبة لمثل هذه المواقف فإن أفضل منهج هو ملف التكرار hashed file. دعنا نفترض لمجرد أن نكون متماسكين أننا ما زلنا نعمل على سجلات الرواتب إلا أننا نهتم الآن بالاتصال المباشر السريع. دعنا نفترض كذلك لنفس الغرض إن كل سجل من السجلات مكون من 100 بايت أو 100 رمز.

على هذا يكون لدينا عدد غير محدد.من السجلات كل منها طوله 100 بايت وكل منها له حقل رئيسي عددي فردي مكون من 5 أرقام. إذا لم يكن هناك قيود على الإطلاق على كيفية تخزين الملف في مجموعة أقراص فكيف يمكننا أن نذهب بالنسبة لتخزين السجلات بالنسبة لأسرع اتصال ممكن؟ شكل 3.16 يبين منظراً لإحدى طرق تخزين البيانات. افرض أننا نختار نقطة بداية اختيارية أو عنوان اختياري على القرص ونضع السجل (إذا كان موجوداً) بقيمة حقله الرئيسي (رقم العامل) 00000 هناك. على ذلك فإننا نضع السجل الذي له قيمة مفتاح 00001 عند النقطة التي تبعد 100 بايت عن نقطة البداية الاختيارية والسجل الذي له قيمة مفتاح 00002 يبدأ بعد 200 بايت من هذه النقطة وهكذا بالنسبة

لكـل سجل (أو في الـواقع السجـلات التي تتواجـد) من السجلات التي لهـا قيمة مفتاح حتى 99999.

Address	Employee Number	Employee Last Name	...
0	00000	Dobbs	...
100	0001	Taylor	...
200	00002	Miller	...
⋮	⋮	⋮	
50.000	00500	Garner	...
⋮		⋮	⋮
9.999.900	99999 .	Monroe	...

← 100 bytes →

شكل 3.16
ملف مباشر عام

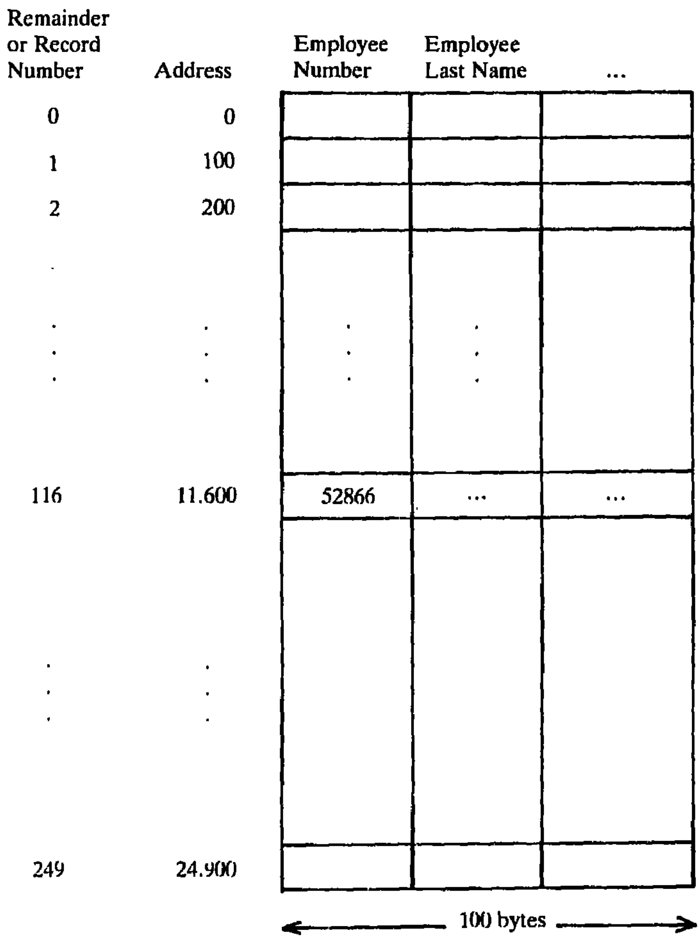
إذا أردنا إيجاد السجل الذي له قيمة مفتاح 00500 فبعد ضرب هذا العدد في طول السجل 100 ننظر عند الموقع الموجود على القرص ويبعد 50,000 بـايت عند نقـطة بداية الملف الاختيارية وهناك نجد السجل الذي نبحث عنـه. ماذا يكـون أبسط؟ أو أسرع؟

ولكن كما هو معتاد عندما يبدو شيء معين بسيطاً فيكون هناك شيء في الجو (أو حيث أننا نتعامل مع أجهزة الكمبيوتر فربما يجب علينا أن نقول أن هناك خطأ بسيط). بالرغم من أننا نستطيع أن نحصل على 100,000 سجل مختلف بمفتاح له خمسة أرقام عشرية فربما لا يمكن أن يكون لدينا هذا العدد من السجلات في الواقع يراد تخزينه. وربما لا يكون هناك سوى 200 عامل فقط في الشركة. النقطة هي أنه لكي نستخدم طريقة بسيطة للتخزين والاسترجاع الذي سبق افتراضه فيجب أن نحجز 10 مليون بايت على القرص كافية لعدد 100,000 سجل بالرغم من أننا نعرف أنه لدينا 200 سجل يراد تخزينها فقط والتي تتطلب 20,000 بايت. مكاناً كبيراً وغير مقبول للتخزين يضيع دون استفادة منه. وهذه هي النقطة. ولكن ما زالت مقدرة مصاحبة قيمة مفتاح بموقع تخزين مغرية جداً للتفكير فيها.

ماذا عن البديل التالي؟ هل نستطيع أن نعمل على إمكانية أن نحصل على مساحة تخزين خاصة بالملف فقط ونحول قيمة الحقل الرئيسي الى قيمة من القيم المتاحة لعناوين التخزين؟ على سبيل المثال هل نستطيع أن نأخذ حقل رئيسي مكون من خمسة أرقام له القيمة 52866 ونحوله الى موقع تخزين نسبي 116 (من 250 موقع تخزين مثلاً)؟

يبين شكل 3.17 تخطيطاً لمنطقة تخزين على قرص نستطيع أن نخزن عليها 250 سجل (مرقمة من 0 الى 249) كل منها طوله 100 بايت. ربما يكون لدينا 200 سجل يراد تخزينها ونريد بالإضافة إلى ذلك ترك مكان للتوسع المتوقع. جرب الطريقة التالية كطريقة مباشرة للتحويل من قيمة مفتاح الى موقع تخزين. اقسم قيمة المفتاح (52866) على عدد مواقع التخزين 250. النتيجة باستخدام حاسبة الجيب هي 211.464. لكن تذكر قواعد المدرسة **وإن عملية القسمة الطويلة وإن** هناك طريقة أخرى لتمثيل النتيجة **وهي 211 والباقي 116. والباقي في عملية** القسمة سوف يكون رقماً صحيحاً دائماً **always يتراوح من 5 الى رقم أقل من** المقام 250 أو إلى 249 في هذه الحالة. **ولكن هذا هو نفس عدد مواقع التخزين** (كل منها طوله 100 بايت) التي **حددناها في المنطقة المبينة في شكل 3.17.** ماذا يحدث إذا أخذنا ببساطة السجل بقيمة مفتاح خاصة لموقع التخزين يحددها باقي عملية قسمة قيمة المفتاح على عدد مواقع التخزين كما سبق توضيحه؟ لكي نسترجع نفس السجل في وقت لاحق فيجب أن نتحرك خلال نفس العملية

ونبحث عن السجل الموجود في الموقع الذي تحدده عملية القسمة. وتسمى هذه الطريقة بطريقة القسمة والباقي divided - remainder method. وهي إحدى الطرق العديدة المسماة طرق التكرار hsahing method والتي يمكن استخدامها في تحويل المفتاح في تخزين ملف التكرار.



شكل 3.17
طريقة القسمة والباقي لملف مباشر.

هناك مشكلة يمكن أن تقابلك. بالتأكيد الاستمرار مع المثال السابق ومع وجود العديد من الأعداد المكونة من خمسة أرقام فيمكن أن يكون باقي قسمة أكثر

من عـدد على 250 هـو 116. ولكن يكون هنـاك عدد واحـد فقط منها (أول عـدد يـظهر لـه هذا البـاقي) يمكن أن يخزن في هـذا الموقـع. وتعرف هـذه المشكلة بأنها مشكلة تضارب collision. قيمتي المفتاح (أو قيم المفتاح) التي تتكرر hash لنفس موقع التخزين تسمى مرادفات synonyms ولا يفقد أي منها فهناك طـرق عديـدة لمعالجة هـذا الموقف. شكـل 3.18 يبين نفس الشكـل الموجـود في شكل 3.17 مـع وجود بعض أماكن تخزين أكثر مضافة بعد الـ 250 موقع مع إضافة مشـير أو حقل

Remainder or Record Number	Address	Employee Number	Employee Last Name	...	Pointer For Synonyms
0	0				
1	110				
2	220				
⋮	⋮	⋮	⋮		
116	12,760	52886	...	...	27,500
⋮	⋮	⋮	⋮		
249	27,390				
250	27,500	37616			27,610
251	27,610	25116			−1
252	27,720				
253	27,830				

← 100 bytes → ← 10 bytes →

شكل 3.18

طريقة القسمة والباقي (مع وجود مرادفات) للملف المباشر

عنوان جديد لكل سجل. وكمثال افرض أننا قمنا بتخزين ثلاثة سجلات لها قيم مفتاح 52866 و 37616 و 25116. هذه السجلات الثلاث يتكرر لها نفس الموقع 116. يخزن أحد السجلات في الموقع 116 والآخر في الموقع 250 والثالث في الموقع 251 لاحظ كيفية تكوين حقل المشير الجديد لسلسلة chain تصل الثلاثة سجلات مع بعضها، إذا حاولت أن تجد السجل الذي له قيمة مفتاح 25116 في الموقع 116 فإنك لا تجد إلا السجل الذي له قيمة مفتاح 52866 هناك. وبعد ذلك فإنك تتبع المشير الموجود في نهاية هذا السجل وذلك الى الموقع 250 (27,500 بايت في الملف). وهناك تجد السجل الذي له قيمة مفتاح 37616 وهذا السجل يشير بدوره الى الموقع 252 والذي تجد فيه أخيراً السجل الذي تبحث عنه والذي له قيمة مفتاح 25116.

وهناك أشياء عديدة أخرى يشار إليها بالنسبة لطرق الاتصال المتكرر.

- لا يوجد تسلسل طبيعي للبيانات المخزنة. فطبيعة الطريقة التكرارية هي نثر السجلات في منطقة التخزين بطريقة تجعل الاتصال المباشر كفؤاً بقدر الإمكان بغض النظر عن أي شيء آخر بما في ذلك التسلسل.

- عبر السنوات طورت العديد من طرق التكرار المختلفة وطريقة القسمة والباقي ما هي إلا إحدى هذه الطرق.

- مقياس الجودة الأكثر أهمية لطريقة التكرار هو كيفية تقليلها لعدد التضاربات. والطريقة الوحيدة لعمل ذلك هي محاولة تخزين مجموعة معينة من السجلات في الواقع بطريقة تكرارية معينة. وأحياناً يمكن عمل تغييرات في مؤشرات الطريقة. مثال ذلك في طريقة القسمة والباقي يكون معروفاً بصفة عامة أنه إذا كان عدد مواقع التخزين وبالتالي المقام عبارة عن عدد أولي أو عدد فردي ولا يمكن قسمته على أي عدد أقل من 20 فيكون هناك على ذلك فرصة أفضل لحدوث تضاربات أقل. وعلى هذا ففي المثال الذي سبق أن فحصناه يمكننا بدلاً من تحديد 250 موقع تخزين ان نحدد عدداً مساوياً للعدد الأولى التالي للعدد 250.

خاتمة Conclusion

ما زلنا في بداية الكتاب إلا أننا حققنا تقدماً هائلاً في تطوير مفاهيم تخزين البيانات. لقد رأينا في الفصل الثاني كيف يمكن ترتيب البت والبايت لتمثيل البيانات وكيفية تنظيم هذه البيانات في ملفات بسيطة. كما ناقشنا كذلك الطريقة التي يمكن بواسطتها وضع هذا البت والبايت التي تمثل البيانات في وسط القرص.

وفي هذا الفصل طورنا فكرة حرجة أخرى لتخزين البيانات وهي أنه هناك بعض التطبيقات التي تتطلب سجلات من ملف لتكون متاحة في تسلسل طبقاً لقيمة آخر الحقول وبعض التطبيقات الأخرى تتطلب أن تكون سجلات الملف متاحة للاتصال المباشر. وطبقاً لأنواع التطبيقات التي تصل للسجلات يجب إعداد بعض الملفات للاسترجاع المتتابع وبعضها للاسترجاع المباشر والبعض الآخر لكل من الاسترجاع المتتابع والاسترجاع المباشر. واستمررنا في هذا الفصل في مناقشة مخططات مختلفة لتنظيم سجلات البيانات لهذه الثلاث فئات لمتطلبات الاسترجاع.

وأثناء بناءنا لقاعدة قوية للعمل منها تركت بساطة (استقلال) الملفات البسيطة الكثير مما نتمناه للكفاءة في بعض مواقف تخزين البيانات الشائعة. وسوف ننظر إلى هذه المشاكل في الفصل الرابع.

مراجع References

Bohl, M., Introduction to IBM Direct Access Storage Devices, Science Research Associates, Chicago, IL, 1981.

Kroenke, D., Database Processing, 2nd ed., Science Research Associates, Chicago, IL, 1983.

Lewis, T.G., and Smith, M. Z., Applying Data Structures, 2nd ed., Houghton Mifflin, Boston, MA, 1982.

Tremblay, J. P., and Sorenson, P. G., An Introduction to Data Structures with Applications, 2nd ed., McGraw-Hill, New York, 1983.

اسئلة وتمارين Questions and Exercises

١ - ما تنظيم الملف؟ وما طريقة الاتصال؟ وماذا يحقق كل منها؟

٢ - ما الملف التتابعي؟ ما هي بعض مميزات وعيوب استخدامه؟

٣ - صف اجراءاً لإضافة سجلات الى ملف تتابعي.

٤ - هل تعتقد أن السجلات الموجودة في ملف تتابعي يمكن تجديدها بدون انتاج ملف رئيسي جديد؟ لماذا ولم لا؟

٥ - صف بكلمات من عندك ماذا يعني الفهرس.

٦ - حيث ان استخدام الفهرس في إيجاد سجل من ملف يشمل بعض العمل أي تنفيذ طريقة بحث الفهرس لماذا يعتبر مثل هذا الاسترجاع مباشراً؟

٧ - صف استخدام الفهرس الخطي البسيط للاسترجاع المباشر لسجل فردي. هل يعني شيء أن يكون الملف في تسلسل طبيعي طبقاً لحقل مفهرس أم لا؟

٨ - صف استخدام الفهرس الخطي البسيط في الاسترجاع التتابعي لكل السجلات الموجودة في ملف. هل يعني شيء أن يكون الملف في تسلسل طبيعي طبقاً لحقل مفهرس أم لا؟

٩ - اعمل فهرس خطي بسيط للملف الموجود في شكل 3.1 مبنياً على حقل معدل الأجر Salary Rate.

١٠ ـ ما معاني الاسترجاع المباشر المعتمد على الفهرس والاسترجاع التتابعي المعتمد على الفهرس إذا لم يكن الحقل المفهرس فردياً؟

١١ ـ ما تنظيم الملف التتابعي المفهرس؟ صف مجموعة من التطبيقات تحتاج إلى أن ينظم الملف بهذه الطريقة.

١٢ ـ ما مميزات الفهرس متعدد المستويات على الفهرس الخطي البسيط؟

١٣ ـ اعتبر فهرس IBM ISAM الموجود في شكل 3.8 افرض أن أحد سجلات الافراد له رقم العامل 77000 وأنه ثالث سجل من أربعة سجلات موجودة على المسار 3 للإسطوانة 11. صف بحثاً مباشراً خلال الفهرس عن هذا السجل.

١٤ ـ اعتبر جزء ملف IBM ISAM المبين في شكل 3.9 اضف السجلات التالية بنفس الترتيب المبين. ماذا يحدث إذا كان مسار السريان الزائد مملوءاً؟

أ ـ 42000 Barber.

ب ـ 39200 Madison.

جـ ـ 41800 Morgan.

١٥ ـ صف آلية التجزئة في تكوين VSAM KSDS.

١٦ ـ قارن تنظيمات الملف IBM ISAM و VSAM KSDS بالنسبة لما يلي:

أ ـ مقدرة الاسترجاع المباشر.

ب ـ مقدرة الاسترجاع التتابعي.

جـ ـ طريقة إدخال سجلات.

د ـ اداء الاسترجاع بعد عدد معتبر من إدخال للسجلات.

١٧ ـ اعتبر جزء الملف VSAM KSDS المبين في شكل 3.13 أضف السجلات التالية بالترتيب المبين. ماذا يحدث إذا كان المسار الحر مملوءاً؟

أ ـ 42000 Barber.

ب ـ 39200 Madison.

جـ ـ 41800 Morgan.

١٨ ـ صف كيف يعمل البحث المباشر في حقل تكراري.

١٩ ـ ما أوجه البحث المتتالي في الملف التكراري؟

٢٠ ـ ما طريقة التكرار؟

٢١ ـ صف كيف يمكن استخدام طريقة القسمة والباقي التكرارية للتأثير على الاسترجاع المباشر لسجل من ملف.

٢٢ ـ ما مميزات وعيوب الملف التكراري بالنسبة للملف المفهرس؟

٢٣ ـ ما التضارب في الملف التكراري؟ وما المرادفات؟ صف أسلوب حل التضارب.

٢٤ ـ ادخل سجلات لها أرقام العمال التالية في ملف الأفراد المبين في شكل 3.18.

أ ـ 32869

ب ـ 00866

٢٥ ـ يوجد مكان في ملف تكراري لعدد 70 سجل طول كل منها 30 بايت. وكل سجل له مكان يناظر 10 بايت لمشير سلسلة تضارب وهناك مكان لعدد 15 سجل إضافي لتخزين السجلات المرادفة.

أ ـ ارسم شكلا يبين هذا الموقف.

ب ـ خزن سجلات لها مفاتيح كل مفتاح يشغل أربعة أرقام على النحو التالي محدداً العنوان المخزن كل سجل فيه.

١ ـ 4000

٢ ـ 5207

٣ ـ 0360

٤ ـ 1410

الفصل الرابع
مزايا مناهج قاعدة البيانات
Benefits of the Data Base Approach

مقدمة Introduction

والآن وبعد أن نظرنا نظرة متفحصة لطرق بناء الملفات الخطية البسيطة وطرق الاتصال بها يحين الوقت لاعتبار مدى جودة هذه الملفات في استخدامها في معالجة البيانات الواقعية وتشغيلها. إلا أنه يجب ملاحظة إحدى النقاط أولاً وتمييزها. فعند الحديث عن الملفات الخطية البسيطة افترضنا أن هذه الملفات موجودة مستقلة تماماً عن بعضها البعض وحتى الآن لم يكن هناك أي سبب لافتراض غير ذلك. وسوف نرى أن معالجة الخليط من هذه الملفات بطرق معينة يمكن أن يقود إلى مفهوم تخزين بيانات أكثر قوة (في الواقع يقود إلى شكل قانوني لقاعدة بيانات). وعلى هذا للتمييز بين الحالتين فسوف نشير من الآن وصاعداً للحالة التي سبق مناقشتها حتى الآن كملفات خطية بسيطة.

وللتأكيد فإن الملفات الخطية البسيطة تكون هي المناسبة تماماً في مواقف معينة. ولكن كما سنرى في هذا الفصل فإنها متخلفة في العديد من الأوجه الأساسية بالنسبة لتطبيقات تشغيل البيانات الأكثر تعقيداً وبالنسبة للبيئات الموجودة حالياً.

دعنا نلقي نظرة على خمسة مناطق مناقشة رئيسية في هذا الصدد وهي ما يلي:

١ ـ خواص معينة لبيئة تشغيل البيانات حالياً.

٢ ـ المشاكل المصاحبة لتكرار البيانات.

٣ ـ المقدرة على تخزين بيانات موجودة في علاقات متعددة.

٤ ـ مواضيع عامة معينة لإدارة البيانات.

٥ ـ مفهوم استقلالية البيانات.

خواص بيئة تشغيل البيانات

Data Processing Environment Characteristics

البيانات كأحد الموارد التي يمكن إدارتها.

في الأيام الأولى لتشغيل البيانات كان يقضي معظم الوقت والتركيز في تطوير التطبيق على البرامج وهذا على عكس التركيز على البيانات وتكوينات البيانات. لقد كانت نظم المكونات مرتفعة الثمن ومحدودة في مدى عملها. وكانت البرمجة أسلوباً جديداً وكان هناك الكثير مما يجب تعلمه عنها ليمكن تحقيق تشغيل مرتفع الكفاءة ولم يكن معروفاً أي نمطية أو آلية جزئية لعملية البرمجة. وفي هذه البيئة كانت معالجة البيانات تأخذ أقصى درجات الاهتمام.

ومع استمرار نمو تشغيل البيانات بدأت تتغير القواعد الأساسية. لقد أصبحت نظم المكونات رخيصة الثمن كما أن تطوير نظم البرامج أخذ شكلاً أكثر نمطية وأكثر ترتيباً structured. كما أنه تكونت صفوف انتظار طويلة لتطبيقات جديدة يراد تنفيذها مما جعل الوقت الكبير اللازم لقضائه في صيانة برامج موجودة بالفعل غير مقبول. وقد تزايد وضوح الطريقة التي نعالج (أو نسيء معالجة) البيانات بها في الماضي ومن أنها كانت أحد العوامل الرئيسية في مجموعة صيانة البرامج التي وجدنا أنفسنا فيها. وقد وجدنا ما يلي:

* إن البيانات مخزنة في أشكال مختلفة في ملفات مختلفة.
* عادة لا تكون البيانات مقسمة على برامج مختلفة تحتاج إليها مما يتسبب في وجود ملفات مكررة.
* عادة لا يكون من السهل استعادة أو ضمان أمن البيانات.
* عادة نكتب البرامج بطريقة بحيث أنه إذا اختلفت طريقة تخزين البيانات فيجب أن يعدل البرامج ليستمر في العمل.

عادة تكون البيانات غير دقيقة أو غير متماسكة أو غير حديثة. فالتغييرات في أي شيء من طرق الاتصال الى جداول الضرائب يلزمها تغييرات في البرمجة.

وفي نفس الوقت ومع تزايد كمية العمل الاستراتيجي والتنفيذي للمؤسسة وتزايد اعتماده على تشغيل البيانات آلياً تواجد حرص متزايد على أن البيانات نفسها ما هي إلا مورد للمؤسسة enterprise resource هام جداً.

الأموال والمعدات الرأسمالية والمخزونات وما شابه ذلك كلها ممتلكات هامة للمؤسسة. لقد طبقنا العديد من أساليب الإدارة عليها (كثير منها يشمل الكمبيوتر بالطبع). والعديد من الناس لديهم الانطباع حالياً بأن البيانات هي أحد الموارد التي يمكن ادارتها كذلك. فبيانات الشركة عن منتجاتها وعمليات التشغيل بها وعملائها ومورديها والعاملين بها ومنافسيها وما إلى ذلك يمكن أن تقدم بالتخزين والاستخدام المناسبين ميزة معنوية للتنافس.

وعند تقاطع كل الأحداث التي سبق ذكرها يوجد السؤال كيف؟ نحن نريد أن ندير البيانات كأحد موارد المؤسسة بينما في نفس الوقت نحسن من معاملتها بالنسبة لتقنية تشغيل البيانات. لقد أصبح من الواضح حتى قبل أن نصل إلى مناقشة التفاصيل التي تأتي فيما بعد في هذا الكتاب أن بعض أنواع النظم المنتظمة لإدارة البيانات كأحد موارد المؤسسة ضرورة. بيئة قاعدة البيانات بنظم برامجها المدعمة وأساليب الإدارة المصاحبة لها سوف تقدم إطاراً لمعاملة البيانات كمورد نمطي يمكن إدارته ويمكن اقتسامه.

النمطية والتخصص

في بيئة تشغيل بيانات تحكمها تطبيقات مستقلة يكون من الصعب عادة حفظ نمطيات في اصطلاحات البرمجة وتخزين البيانات. فيميل المبرمجون إلى أن يسلكوا طرقهم الخاصة وبصفة خاصة عندما لا تكون البرامج الأخرى معتمدة اعتماداً مباشراً على عملهم. ولسوء الحظ فإن القصور في النمطية يجعل من الصعب على أي شخص حتى بالنسبة للمبرمج الأصلي أن يفهم البرنامج فهماً جيداً يجعله قادراً على عمل التغييرات اللازمة على البرنامج فيما بعد.

اقتسام التطبيقات المختلفة لملفات البيانات واستخدام عبارات اتصال

ووصف متماسك للبيانات في البرنامج في بيئة قاعدة البيانات كل ذلك يفرض وجود نظرة حريصة ونمطية للعملية الداخلية لتطوير التطبيق. والآن فما يفعله أي مبرمج يمكن أن يؤثر على الآخرين بالنسبة الى دقة البيانات وتجديد الجدولة والأداء على سبيل المثال. والنمطية في أساليب البرمجة وفي مخططات التخزين يمكن أن تخدم في جعل البيئة لا ينتج عنها اخطاء كثيرة فقط.

بالإضافة إلى ذلك تقود بيئة قاعدة البيانات الى مستويات جديدة من التخصص في العمل وهي في واقع الأمر صورة أخرى من النمطية. وبصورة تقليدية فإن وظيفة إدارة البيانات تشكل لتكون مسؤولة عن البيانات. وداخل الوظيفة يوجد أفراد متخصصون في تصميم قواعد البيانات وأمنها وإعداد نسخ احتياطية منها واستعادتها وأدائها وما إلى ذلك. ومثل التعهد للجديد تتزايد البيئة الموجهة للبيانات وتقوم مجموعة عمليات وبرمجة التطبيقات بتطوير اعتمادية على وظيفة ادارة البيانات ويتزايد كم عمليات التنفيذ وتطوير التطبيقات الموجهة ناحية البيانات.

ويمكن أن يشمل ذلك تجميع البيانات والتأكد من صحتها وتوجيهها وتصميم قاعدة البيانات وتوجيه تكامل البيانات وأمنها وعمل النسخ الاحتياطية وتجديد البيانات أي حراسة دقيقة لبيانات المؤسسة. ولا يحدث ذلك في يوم وليلة ولكنه يحدث تدريجياً. ولكن مرة أخرى فإن هذا النوع من النمطية - الأفراد والنمطية الوظيفية في هذه الحالة - يمكن أن يقود الى بيئة أكثر كفاءة وأقل أخطاء فقط.

تكرار البيانات Redundant Data

دعنا نلقي نظرة على مشكلة تكرار البيانات وما يصاحبها من سلامة البيانات. أولاً سنعتبر المشكلة في مضمون ملف واحد ثم في مضمون عدة ملفات.

التكرار داخل الملف الواحد

إفرض أن هناك شركة تجارية وبالطبع يكون لها طريقة لحفظ تتبع عملائها وبائعيها. يبين شكل 4.1 ملفين خطيين مستقلين أحدهما للبائعين والآخر

للعملاء. ملف البائعين له مفتاح رئيسي هو حقل رقم البائع SALESPERSON NUMBER ويشمل حقل اسم البائع SALESPERSON NAME كأحد حقوله. أما ملف العملاء فله مفتاح رئيسي (فردي كذلك) هو رقم العميل CUS-TOMER NUMBER وبه حقل رقم البائع SALESPERSON NUMBER كأحد حقوله وهو رقم البائع المحدد لرقم حساب هذا العميل. وهناك سجل واحد لكل بائع في ملف البائعين. والسجل يحتوي على بيانات تحفظ عن البائع نفسه مثل رقمه واسمه وعنوان منزله ورقم الهاتف بمنزله وما إلى ذلك. وهناك موقف شبيه للبيانات بالنسبة لكل عميل في ملف العملاء. ويقال على هذا النوع من أنواع تخزين البيانات بأنه ليس فيه تكرار nonredundant لأن البيانات التفصيلية عن كل كينونة مخزنة مرة واحدة فقط، وهذا ما يجب أن يحدث.

SALESPERSON NUMBER	...	SALESPERSON Name	...
137		Baker	
186		Adams	
204		Dickens	
361		Carlyle	
. . .		. . .	

(a)

CUSTOMER NUMBER	...	SALESPERSON NUMBER	...
0121		137	
0839		186	
0933		137	
1047		137	
1525		361	
1700		361	
1826		137	
2198		204	
2267		186	
. . .		. . .	

(b)

شكل 4.1
ملفات منفصلة
(ا) البائعين (ب) العملاء

إذا حضر شخص معين وأراد أن يعرف اسم البائع المعرف برقم بائع معين (مثل 186) فإنه ينظر في سجل هذا البائع في ملف البائعين. شكل 4.1 a يستخدم

رقم البائع كمفتاح يمكن أن يقرأ منه اسم البائع على أنه Adams وبالمثل إذا أراد شخص آخر أن يعرف رقم البائع المسؤول عن حساب أحد العملاء فإنه يبحث في ملف العملاء في شكل 4.1 b في سجل هذا العميل مستخدماً رقم العميل كمفتاح. هذا شيء جيد حتى الآن.

والآن ماذا يحدث إذا أراد أحد الأشخاص أن يعرف اسم البائع الذي يخدم حساب عميل معين ومعرف برقم العميل. هل يمكن الحصول على المعلومات من أحد الملفات فقط في شكل 4.1؟ الواضح أن هذا لا يمكن حيث أن معلومات العميل موجودة في ملف العملاء فقط واسم البائع موجود في ملف البائعين فقط. والطريقة الوحيدة للإجابة على هذا السؤال هي النظر في سجل العميل أولاً في ملف العملاء وإيجاد رقم البائع من هذا الحساب ثم استخدام هذا الرقم في إيجاد سجل البائع في ملف البائعين وأخيراً إيجاد الاسم من هذا السجل. مثال ذلك في شكل 4.1 إذا كان رقم العميل هو 1525 فيمكننا الاتصال بالسجل الخامس من ملف العملاء ومعرفة أن البائع المسؤول هو البائع رقم 361. بعد ذلك يتم الاتصال بسجل البائع رقم 361 في ملف البائعين وإيجاد الاسم على أنه Carlyle. هذا النوع من الأعداد طبقاً للعميل والأوامر المتعددة والاتصال المتعدد يتسبب في حدوث أخطاء بالنسبة للبرمجة كما أنه مكلف بالنسبة للتنفيذ ويعتمد على نظم البرامج المستخدمة والتي يمكن أن تعد كل منها على حدة بالنسبة لخليط الاتصال هذا.

ولكن إذا كنا نعرف أننا سوف نجري مثل هذا البحث فلماذا نقوم بتجزئة البيانات في ملفين منذ البداية. إذا وجدت البيانات كلها في ملف واحد بطريقة ذات معنى فلا يكون هناك اتصال مكلف به العديد من الأوامر والعديد من الاتصال بالملفات. يبين شكل 4.2 نفس البيانات ممثلة في ملف واحد. لاحظ أننا نبين البائع (برقمه) المحدد لعميل معين كما فعلنا تماماً في ملفى شكل 4.1 ولكن يوجد اسم البائع كذلك الآن ويمكننا أن نجيب على السؤلين الأصليين البسيطين الموجودين بالنسبة لملفى شكل 4.1. ولكن بالإضافة الى ذلك يمكننا الآن الإجابة على السؤال الأصعب متعدد الملفات السابق : ما اسم البائع الخاص برقم عميل معين؟ وذلك في ملف واحد وأمر واحد في شكل 4.2. حسناً، في الملف الموجود في شكل 4.1 a يوجد سجل واحد لكل بائع وكل سجل يحتوي على بيانات خاصة

بهذا البائع مرة واحـدة. ويكون تحـديد خـدمات البـائع لحسـاب معين عبـارة عن توصيل رقم البائع بسجل فردي يصف العميل في شكل 4.1 b.

SALESPERSON NUMBER	CUSTOMER NUMBER	SALESPERSON NAME	...
137	0121	Baker	
137	0933	Baker	
137	1047	Baker	
137	1826	Baker	
186	0839	Adams	
186	2267	Adams	
204	2198	Dickens	
361	1525	Carlyle	
361	1700	Carlyle	
	⋮	⋮	

شكل 4.2
ملف مركب

ولكن في الملف المـركب في شكـل 4.2 حيث دمجت كـل الأنـواع المختلفـة للمعلومات أساسـاً فيجب أن تكرر البيـانات الخـاصة بـالبائـع لكل حسـاب يقوم بخدمته. ويمكن أن يظهر أحـد البائعـين في العديـد من السجلات في هـذا الملف وهذا لا معنى له من وجهة النظر المنطقية أو من وجهة نظر الاسـترجاع. ومثـال ذلك اسم البائع لعميل واحد يخدمه هـذا البائـع ولايخدم آخـر. ولاتمام ذلـك يجب أن تتكـرر البيانـات في كل مكـان مناسب. وعـلى هـذا فبينـما اسم Carlyle في ملفى شكل 4.1 يصاحب رقم البائع 361 مرة واحدة فـإن نفس اسم Carlyle يظهـر في شكل 4.2 مع رقم البائع 361 بعدد العملاء الذين يخدمهم هذا البائع.

يتسبب هذا الموقف في عدد من المشاكل. أحد الأشياء هي أنه يشغل مكانـاً أكثر في التخزين. (حاول أن تضيف حقل عنوان المنزل المكون من 40 بايت وانظر كيف يؤثر ذلك على التخزين والذي قد يزيد أضعافاً مضاعفة عما هو الحـال عليه

الآن). وهناك شيء آخر فإن هذا الموقف غير مريح عند عمل التجديد. إذا غير Carlyle اسمه (هذا شيء معتاد بالنسبة للبائعات في المجتمعات الغربية عند زواجهن حيث يتغير اسم عائلتهن من اسم عائلة الأب إلى اسم عائلة الزوج: المترجم) فيجب أن يجري التغيير العديد من المرات. ويمكن أن يكون الوقت اللازم لعمل ذلك مكلفاً جداً وبصفة خاصة إذا كانت السجلات المقصودة مبعثرة على القرص كله. كما أنه هناك شك مزعج أيضاً بأن كل ما يحدث يجري تجديده وذلك لأي سبب من أسباب أخطاء البرمجة البسيطة الى وقت التنفيذ وحدوث فشل في النظام. وعندما يحدث ذلك فإنه يسمى بمشكلة السلامة integrity.

ويبدو أنه هناك معاملة تبادل إلا أن التبادلات نادرة في مجال الكمبيوتر. 4.2 لا يوجد تكرار للبيانات (على الأقل بالنسبة إلى الحقول غير الرئيسية). إلا أن البحث من النوع المطلوب يلزمه اتصال متعدد الأوامر ومتعدد الملفات. أما في الملف المركب في شكل 4.2 فقد ألغينا الحاجة إلى اتصال متعدد الأوامر ومتعدد الملفات لهذا النوع من البحث إلا أننا قدمنا تكراراً عبر الحقول غير الرئيسية. أي من هذين الموقفين يكون متساهلاً للأسباب التي سبق ذكرها.

والسؤال هو هل يمكن أن يكون لدينا الكعكة ونأكلها كذلك؟ وهذا يحضرنا إلى أحد السمات الرئيسية لبيئة قاعدة البيانات فنظام إدارة البيانات المتكامل integrated، وهو نظام قاعدة بيانات database حقيقي، هو نظام يمكن أن تحفظ فيه البيانات دون تكرار (بمفهوم ملفات شكل 4.1). وفي نفس الوقت فإن البحث الذي يتطلب خليطاً من أنواع مختلفة من البيانات (مثل البحث الذي نريده) يمكن أن يحدد في أمر واحد من أعلى مستوى لسطح بيني للبرمجة (كما لو كان الاتصال يحدث للملف الموجود في شكل 4.2). وأي نظام ليس لديه هذه الخاصية يجب ألا يسمى نفسه بأنه نظام «قاعدة بيانات» في الواقع فنظام «إدارة البيانات» يجب أن يكون أكثر دقة.

هناك مناهج عديدة لتصميم نظام قاعدة بيانات حقيقي يدور حول تكوينات تخزينات عديدة وأوقات مختلفة وطرق خلط mixing قطع بيانات مختلفة مع بعضها. وسوف نستكشفها فيما بعد.

وهناك ميزة أخرى من المنهج المتكامل وهي التشجيع الذي يقدمه في إيجاد مستفيدين جدد للبيانات الموجودة في قاعدة البيانات. وثراء وجود اجزاء مختلفة عديدة لبيانات المنظمة متصلة ببعضها يقود الى امكانيات استخدام أكثر مما لو كانت هذه الأجزاء مخزنة منفصلة عن بعضها.

التكرار عبر العديد من الملفات

هناك نوع آخر من التكرار وهو بالرغم من حتمية وجوده عن النوع الذي سبق ذكره ليس بالضرورة أسهل في حفظه وهذا لعدة أسباب.

عادة تتكرر الحاجة الى بيانات معينة في أقسام مختلفة في إحدى المؤسسات أثناء أدائها عملها اليومي. مثال ذلك قد يكون مطلوب اسم العميل وعنوانه في قسم المبيعات وفي قسم حسابات المدينين وفي قسم الديون. كما يكون هناك بصفة عامة قطع معلومات أخرى تشمل العملاء الذين يحتاج إليهم قسم واحد أو قسمين من أقسام المؤسسة. وحل هذا الاجتياح المتعدد هو التكرار عادة. فقسم المبيعات لديه ملفاته الخاصة والتي تحتوي ضمن ما تحتوي على اسم العميل وعنوانه ونفس الشيء يحدث بالنسبة لقسم حسابات المدينين وقسم الديون.

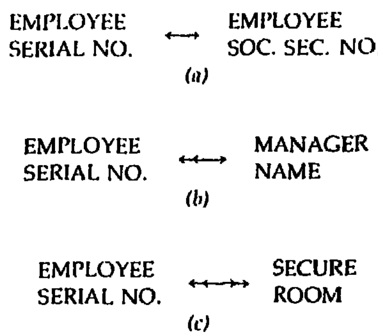
وبينما يكون الصحيح أن هذا يجعل الأقسام لا تتنافس على البيانات إلا أنه يفتح عليه من علب الديدان وهي حافلة بالذكريات بالمشاكل التي نقابلها في تكرار البيانات في ملف فردي. أول كل شيء تخزين قطعة من البيانات ثلاثة مرات يأخذ مكاناً يعادل ثلاثة أمثال المكان لتخزينها مرة واحدة. بالإضافة إلى ذلك ففي كل مرة يجب أن تجدد البيانات يجب أن تتغير في كل موقع تخزين كذلك. وهذا يمثل مضيعة لوقت التشغيل ويتسبب في مشاكل في التحكم والمراقبة ويقود الى مشاكل في السلامة إذا لم تجدد كل البيانات في نفس الوقت. الأكثر من ذلك يظهر سؤال خاص بمن المسئول عن تجديد البيانات في كل هذه الملفات المختلفة والتي تكون مملوكة للعديد من الأقسام المستقلة من أقسام المؤسسة.

أليس من الأفضل أن تخزن هذه البيانات مرة واحدة فقط بطريقة تجعل الاتصال بها ميسراً لكل المجموعات المختلفة التي تحتاج إليها؟ وبهذه الطريقة يوجد مجموعة واحدة مسؤولة عن تجديدها كما يوجد مكان ضائع أقل وتقل مشاكل السلامة وما إلى ذلك.

يشمل منهج قاعدة البيانات مفهوم قاعدة البيانات المشتركة التي يقتسمها الآخرون ويكون هناك ادارة لمراقبتها ومسؤولية لتجديدها.

العلاقات المتعددة Multiple Relationships

لقد ناقشنا في الفصل الثاني الطرق المختلفة التي يمكن أن ترتبط بها الكينونات ببعضها البعض بالنسبة للمصاحبة الفردية أو المصاحبة المتعددة والعلاقات من واحد لواحد أو من واحد للعديد أو من العديد للعديد شكل 4.3. والسؤال الذي يجب أن نسأله الآن هو «ما درجة سهولة تخزين بيانات تشمل النوعين من المصاحبة والثلاثة أنواع من العلاقات في ملفات خطية مستقلة»؟



شكل 4.3
العلاقات: (ا) علاقة من واحد إلى واحد
(ب) علاقة من واحد إلى العديد
(جـ) علاقة من العديد إلى العديد

في كلمات عامة تعالج المصاحبات الفردية والعلاقات من واحد إلى واحد جيداً بمثل هذه الملفات. وبعد كل ذلك يمثل حقل رئيسي وحقل غير رئيسي بسيط مثل هذه الحالة شكل 4.4. ولكن بالنسبة للمصاحبات المتعددة والعلاقات الناتجة منها فتترك الملفات البسيطة شيئاً (لا يوجد بها) مرغوباً فيه.

EMPLOYEE SERIAL NO	SOCIAL SECURITY NO	MANAGER	...
1234	111-000-2222	Smith	
1572	111-22-3333	Jones	
2186	222-11-5432	Jones	
4522	522-52-5221	Smith	
4991	333-55-9999	Doe	
5283	123-45-6789	Smith	
5542	654-32-1234	Smith	
⋮	⋮	⋮	

شكل 4.4
ملف بسيط

يمكن تمثيل مصاحبة متعددة أفقياً بسجلات متغيرة الطول شكل 4.5 a. ولسوء الحظ يمكن أن يسبب ذلك مشاكل في منطقة البرنامج وفي إدارة المكان. عادة إذا كانت اتصالات المبرمج بالبيانات عبارة عن سجل واحد من الملف فيكون هناك اجراء يؤخذ على قيمة فرّدية في أحد حقوله أو في أكثر من حقل من حقوله. في شكل 4.4 يمكن للمبرمج أن يقرأ سجل الموظف رقم 2186 ويمكنه أن ينفذ بعض الاجراءات برقم الضمان الاجتماعي الفردي الموجود فيه. ولكن ماذا إذا اتصل المبرمج بسجل المدير Smith في شكل 4.5 a بهدف العمل على أعداد متسلسلة للأفراد العاملين تحت رئاسة هذا المدير؟ يجب عليه في هذه الحالة وتحت تحكم البرنامج أن يدير قائمة بالأرقام المسلسلة للعاملين وينفذ الاجراء المطلوب عليها. وهذا يجعل البرنامج منتجاً للخطأ أكثر. كما أن العمل مع سجلات متغيرة الطول يمكن أن يكون عملاً ليس سهلاً من وجهة نظر ادارة المكان. فإذا أضيف عامل جديد إلى سجل المدير في ملف شكل 4.5 a فيمكن أن يزيد الحجم الجديد الأكبر للسجل عن الموقع الأساسي المخصص له على القرص الذي أتى منه. ووضعه في مكان آخر يمكن أن يسبب مشاكل في الأداء في الاسترجاع فيما بعد.

MANAGER	EMPLOYEE SERIAL NO'S	. . .
Doe	4991	
Jones	1572 2186	
Smith	1234 4522 5283 5542	
. . .		

شكل 4.5a

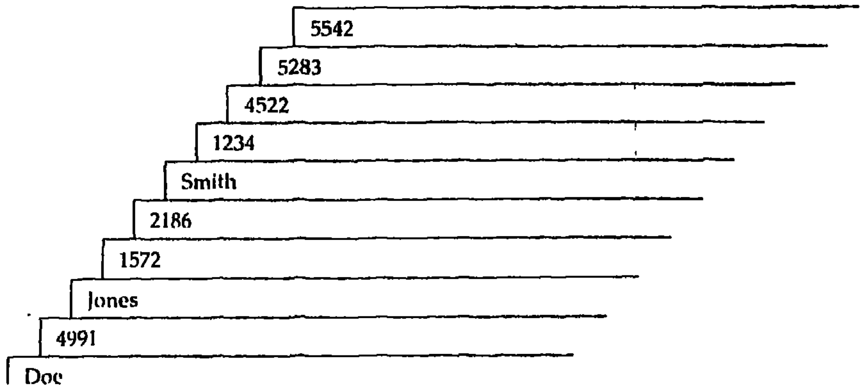
تمثيل علاقات متعددة: تمثيل أفقي

MANAGER	EMPLOYEE SERIAL NO	. . .
Doe	4991	
Jones	1572	
Jones	2186	
Smith	1234	
Smith	4522	
Smith	5283	
Smith	5542	
. . .	. . .	

شكل 4.5b

تمثيل علاقات متعددة: تمثيل رأس

والمصاحبة المتعددة يمكن أن تمثل رأسياً وذلك بتكرار أحد الحاضرين مرة لكل من الحاضرين المرتبطين به، شكل 4.5 b. إلا ان هذا يقدم صورة من صور التكرار: يظهر اسم Smith مرة لكل من العاملين تحت رئاسة هذا المدير (وسوف يظهر أنه عند التشكيل بتكوين علاقي relational فإن الملفات من نوع الملف الموجود في شكل 4.5 b سوف تعتبر كقاعدة لحل قاعدة البيانات القانونية. وتحت



شكل 4.5c

تمثيل علاقة معقولة : سجلات منتشرة

هذه الظروف فإن نوع التكرار الموجود في شكل 4.5 b والخاص بالحقول الرئيسة لكينونتين مشمولتين في مصاحبة متعددة يكون مقبولاً. يوجد المزيد عن ذلك في فصلي 8و 9).

يمكن تمثيل المصاحبة المتعددة بأنواع سجلات منتشرة هنا وهناك شكل 4.5 c. إلا أن هذا يتطلب بعض وحدات نظم البرامج المنسقة للحفاظ على سلامة ما هما ملفين مختلفين بالفعل (ممثلين بنوعي سجلات مختلفين) وبين علاقة واحد (اسم المدير) الى العديد (رقم العامل) المبينة في شكل 4.5 c. مرة أخرى ا لعودة الى كل المشاكل تشكل في الواقع نوعاً من حل قاعدة البيانات (والذي سوف يناقش فيها بعد).

أحد مقدرات نظم إدارة قاعدة البيانات هي أنها تقدم الوسائل الطبيعية السهلة لكل أنواع التطبيقات لتمثيل المصاحبات المتعددة ووجود علاقات واحد إلى العديد وعلاقات العديد الى العديد.

مواضيع عامة لإدارة البيانات General Data Management Issues

هناك اهتمامات عامة بالطريقة التي ندير بها البيانات التي يجب أن تعتبر في أي بيئة تشغيل بيانات بغض النظر عن تكوين الملف المستخدم. هذه الاهتمامات داخلية بالنسبة لتخزين البيانات وإدارتها واسترجاعها. وهي ما يلي:

الأمن

الاحتياطي والاستعادة recovery.

التزامن Concurrency
القابلية لأعمال المراجعة الحسابية auditability .

الأمن

هناك في الواقع عدة أوجه للأمن في بيئة تشغيل البيانات وبعضها فقط يتعامل مباشرة وفورياً مع الاتصال بالبيانات وهو ما يهمنا في هذا الكتاب. مثال ذلك لمنع شخص غير مسموح له من الاتصال بالنظام نفسه فعادة ما تستخدم كلمات مرور خاصة (وكلمات سرية كما يطلق عليها هذا الاسم عادة). وفي الأجنحة المختلفة والأبواب الكبيرة في بعض الأحيان يوجد حراس مسلحون لحماية أجهزة الكمبيوتر من أن يدمرها أي شخص. وهناك أوجه عديدة للأمن وهي مرتبطة بصفة خاصة بالبيانات وبالتالي فهي تهمنا هنا. النقطتين الاساسيتين هما منع الناس من رؤية بيانات غير مسموح لهم الاطلاع عليها ومنعهم من تغيير بيانات غير مسموح لهم أن يقوموا بتغييرها.

ومن المؤكد أنه من الممكن تصميم نظم وبنائها يمكنها أن تحمي البيانات الموجودة في ملفات خطية مستقلة. والمشكلة هي أنه بدون موافقة كل شخص على استخدام مثل هذه النظم فسوف يترك المبرمج مع المعدات إما ليحذف مثل هذه الاعتبارات للأمن أو لإعادة ابتكار العجلة لكل تطبيق.

الاحتياطي والاستعادة:

افرض أن إحدى الحوادث أو إحدى الكوارث الطبيعية أتلفت أحد مجموعات أقراصك. أو افرض أنه اكتشف أحد الأخطاء في المدخلات وأنه حدث منذ انقضاء 10 دقائق وفي بيئة الوقت الحقيقي أثر هذا الخطأ على البيانات.

من الواضح أنه من الأفضل وجود أكثر من نسخة واحدة لكل ملف من ملفات البيانات. أكثر من ذلك يجب أن تحفظ النسخ في مبان مختلفة أو حتى في مدن مختلفة لمنع الكوارث من تدمير كل النسخ الخاصة بالبيانات. عملية نسخ الملف والتي يجب أن تحدث بصورة دورية لتعكس التغييرات في البيانات تعرف بأنها احتياطي back up أو backing up. وعملية استخدام نسخة الملف بالإضافة إلى بيانات أخرى لتصحيح أي من أنواع المشاكل التي سبق ذكرهاتعرف بأنها

استعادة.

وهنا مرة أخرى كما في حالة الأمن من المؤكد أن من الممكن عمل نظم لعمل الاحتياطي والاستعادة للملفات خطية مستقلة مبنية على تشغيل البيانات. ولكن مرة أخرى هذا يعني أن كل فرد في البيئة يجب أن يستخدم النظام وهذا عادة ما يكون عكس الطبيعة المستقلة للتطبيقات في مثل هذه البيئات.

التزامن:

في تطبيق المستفيدين المتعددين مثل نظام حجز مقاعد الطائرات ومكاتب الحجز العديدة يمكن أن تظهر مشكلة خاصة بغض النظر على هيكل الملف المستخدم وتشمل تغيير قيم الحقول في السجلات. عندما يحاول مستفيدان اثنان تجديد نفس السجل في نفس الوقت (أو في نفس اللحظة كما يسمح نظام البرمجة المتعددة) فيكون عندهما طريقة مزعجة جداً للتداخل مع بعضهما بحيث أن أحد هذين المستفيدين يهمل التجديد الذي يجريه.

والحل الشائع لهذه المشكلة يشمل أسلوب يسمى الإغلاق lockout. ولسوء الحظ وكما يحدث عادة فإن استخدام هذا الأسلوب في حد ذاته يحدث مشاكل أخرى لم يسبق لها الوجود والتي يجب أن تعالج.

مرة أخرى هناك اهتمام خاص: من أو ما هو نظام البرامج المسؤول عن تنفيذ هذه الوحدات التي تجري الحماية؟

القابلية للمراجعة المحاسبية

كلما ازدادت بيانات المؤسسة المخزنة والتي يجري تشغيلها آلياً كلما ازدادت أهمية التركيز على وظيفة المراجعة المحاسبية. وهناك أسباب عديدة للإدارة والمحاسبين وغيرهم للسؤال عن من اتصل بالبيانات أو من أجرى تغييرات على البيانات وما البيانات المقصودة وفي أي وقت حدث ذلك وما القيم الجديدة والقديمة للبيانات؟

والوسيلة الأولية لمراجع الحسابات هي محاولة المراجعة. يعد سجل ثابت لطبيعة التغييرات التي تجري على البيانات. ولكن هنا كما هو الحال في الثلاثة مواقف التي سبق ذكرها من أو ما هي المعدات التي يجب أن تكون مسؤولة عن وظيفة المراجعة؟

الملفات المستقلة ومواضيع عامة لإدارة البيانات :

في الأربع حالات كلها كان المفهوم الأساسي هو أن وجود طريقة مشتركة لمعالجة هذه المناطق مبنية في نظام ادارة قاعدة البيانات DBMS يكون مناسباً. وهذا يعني أنه عند كتابة برنامج تطبيق جديد يمكننا تجنب التكاليف التي يتحملها المبرمج في كتابة ما يخص الأمن والاستعادة وما إلى ذلك من البداية أو محاولته استخدام مجموعات نظم برامج متخصصة لكنها غير متصلة بهذه الوظائف. وهذا يعني أننا نستطيع أن نتأكد من أن الأجزاء تعمل حيث أنها مكونات نمطية من نظام ادارة قاعدة البيانات DBMS ثم اختبارها. وهذا يعني أن الوظائف نمطية لكل برامج التطبيقات في البيئة وهذا يقود إلى إدارة أسهل لها واقتصاديات أفضل في تحديد وتدريب الأفراد الذين يكونوا مسؤولين عن هذه الأوجه. هذا النوع من المشاركة عبارة عن صفة رسمية لمنهج قاعدة البيانات.

استقلال البيانات Data Independence

في بيئة تشغيل بيانات الملفات الخطية المستقلة تؤخذ القرارات التي تشمل طريقة كتابة برنامج التطبيق عادة في انسجام مع اختيار تنظيم الملف وطريقة الاتصال. ويعتمد منطق البرنامج نفسه على طريقة تخزين البيانات. وفي الواقع عادة ما يكون الاعتماد dependence قوياً لدرجة أنه لأي سبب من الأسباب عندما تتغير خواص تخزين البيانات في المستقبل يجب أن يعدل البرنامج نفسه وعادة ما يكون هذا التعديل كبيراً. وهذه خاصية غير مرغوب فيها بسبب الوقت والجهد اللازمين لذلك. وعملياً فإن تكوينات التخزين يجب أن تتغير لتعكس الأساليب المعدلة للتخزين ونظم المكونات الجديدة ومحاولات المشاركة في البيانات وضبط الأداء وهذه ما هي إلا أسباب قلة من أسباب كثيرة. وأحد أسباب الوقت الكبير الذي يقضي في صيانة البرنامج هو المستوى التقليدي للاعتمادية بين البرامج والبيانات.

وما نود أن نكون قادرين على عمله هو فصل اعتبارات التخزين والاتصال عن البرمجة بأكبر قدر ممكن. ونظم إدارة قواعد البيانات DBMS الحالية نجحت الى حد ما في هذا الاتجاه في حل بعض أوجه هذه المشكلة.

خاتمة : Conclusion

يجب إعادة النظر إلى استخدام الملفات الخطية المستقلة التي ظهرت مع بداية تشغيل البيانات آلياً وذلك كنقطة بداية وكمقياس للفجوة الموجودة بينها وبين قواعد البيانات. فمن الواضح أن طريقة الملفات الخطية المستقلة لتخزين البيانات المبرمجة مع جو من التطبيقات الفردية التي تنفذ في فراغ تحمل بين طياتها العديد من الآثار الجانبية.

الحل كما سنرى أكثر وأكثر هو منهج قاعدة البيانات.

مراجع: References

Date, C. J., An Introduction to Database Systems, 3rd., vol. 1, Addison - Wesley, Reading, MA, 1981

King, J. M., Evaluating Data Base Management Systems, Van Nostrand Reinhold, New York, 1981.

Kroenke, D., Database Processing, 2nd ed., Science Research Associates, Chicago, IL, 1983.

Martin, J., Computer Data - Base Organization, 2nd ed., Prentice - Hall, Englewood Cliffs, NJ, 1977.

Ross, R.G., Data Systems, AMACOM, New York, 1978.

اسئلة وتمارين: Questions and Exercises

١ - هل يجب أن تعتبر البيانات مورداً حقيقياً للمؤسسة؟ لماذا ولم لا؟ قارن وضاهي البيانات مع الموارد الأخرى للمؤسسة (رأس المال والمباني والمعدات والأفراد وما إلى ذلك). وذلك بالنسبة للأهمية وللقيمة الجوهرية وطرق الاستخدام.

٢ - دافع عن أو هاجم العبارة التالية: (البيانات هي أهم مورد من موارد المؤسسة لأنها تضف الموارد الأخرى).

٣ - كيف تغير بيئة قاعدة البيانات من طريقة معاملتنا للبيانات والتفكير فيها؟

٤ ـ ما نوعي تكرار البيانات وما المشاكل التي تظهر في بيئة تشغيل البيانات بسببهما؟

٥ ـ اذكر مثالاً يوضح مشكلة تكرار البيانات في ملف.

٦ ـ صف تبادلاً واضحاً بين تكرار البيانات وسلامة البيانات في ملفات خطية بسيطة.

٧ ـ بكلمات من عندك صف الجودة الرئيسية لنظام إدارة قاعدة بيانات DBMS يوضح في جانب آخر من جوانب نظم معالجة البيانات الأخرى.

٨ ـ صف الموقف الذي تحفظ فيه البيانات متكررة في ملفات متعددة.

٩ ـ هل تعتقد أن مشكلة تكرار البيانات في ملف فردي أكثر خطورة أم أقل خطورة أم أنها تمثل نفس الخطورة مثل مشكلة تكرار البيانات في ملفات متعددة؟

١٠ ـ اختر جزء وظيفي معين من أحد المؤسسات وصف عدة علاقات من نوع واحد لواحد وواحد للعديد والعديد للعديد بين كينونات هذه الوظيفة.

١١ ـ بكلمات من عندك صف المشكلة الموجودة في محاولة معالجة المصاحبات المتعددة في الملفات الخطية البسيطة.

١٢ ـ ما مدى أهمية المقدرة على معاملة مصاحبات متعددة لتطبيقات تشغيل البيانات؟ لماذا؟

١٣ ـ صف المواضيع العامة لإدارة البيانات التي ذكرت في هذا الفصل. ما مدى أهميتها لنظام ادارة قاعدة بيانات ليكون قادراً على معالجتها؟

١٤ ـ ما استقلالية البيانات؟ ولم تكون مرغوب فيها؟

الفصل الخامس
خواص نظام إدارة قواعد البيانات
Database Management Systems Characteristics

طبيعة نظام إدارة قاعدة البيانات

The Nature of Database Management system

استيعاب مفهوم نظام ادارة قاعـدة البيانـات يتطلب بعض التفكـير في فكرة الرجل الوسيط middleman فهناك العديد من الأمثلة للرجل الوسيط في العالم مع عدد من الاختلافات.

أعتبر على سبيـل المثال فكـرة شراء سلع من شركة من الشركـات التي تبيع عن طريق البريد (هذه الشركات منتشرة جداً في الدول الغربية حيث يرسـل إليها طلب الشراء من العميل بالبريد أو عبر الهاتف لترسل لـه السلع عن طريق الـبريد ويتم الدفع عن طريق البريد أو البنك أو بـأي طريقـة تحددهـا الشركة سـواء دفعاً فـورياً أو يسبق الاستـلام أو بالتقسيط: المـترجم)، شكل 5.1. في أحـد الأطراف يوجد العميـل الذي يـريد شراء بعض حـاجياتـه وفي الطرف الآخـر يوجـد مخزن الشركة الموجـودة به السلع. وفي المنتصف يـوجد خليط من كتـالوج يصف السلع والناس وتسهيلات قسم الكتالوج بالشركة.

لا يستطيع العميل أن يصل إلى مخزن الشركة دون المـرور عبر قسم الأوامـر من الكتالوج. وهـذه نقطة مهمـة لأنها تسمح للشركـة بأن تحتفظ بمـراقبة مخـزون ومحاسبة دقيقة على سلعها الموجودة في المخزن. والسماح للعملاء بـالاتصال المبـاشر بالمخزن قد ينتج عنه مشاكل مراقبـة خطيرة. والـتركيز عـلى الكتالـوج يظهـر نقطة مهمة أخرى بالنسبة لهذا الموقف. فالكتالوج في حد ذاتـه ليس سلعة لكنـه وصف للسلع. يمكن أن يكون هناك 300 نـوع من بكرات الصيـد في المخزن ونـظر لأنها

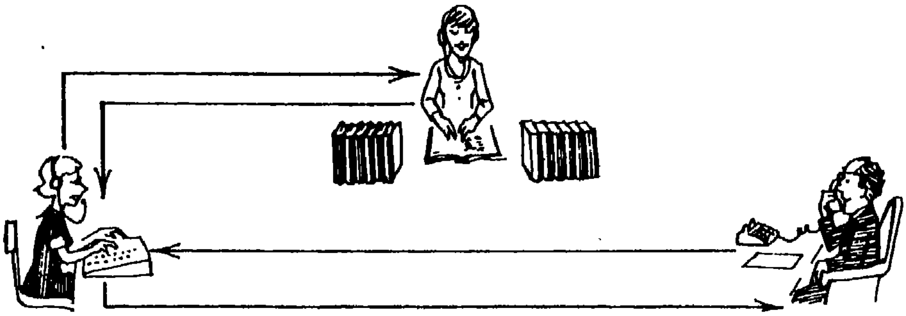
كلها متطابقة فلا يوجد إلا وصف واحد لها فقط مع ذكر أنه يوجد 300 نوع منها فقط وهذا يكفي لوصف ما هو موجود بالمخزن. وكل من العميل وقسم الأوامر بالكتالوج يستخدم الكتالوج كوسيلة لمعرفة شيء معين عن بكرات الصيد دون الدخول الفعلي للمخزن ولمسه على الرف. وهذا مفهوم أيضاً. ربما يضع العميل أمراً لقسم الأوامر بالكتالوج ويفحص القسم الكتالوج ومعلومات إرشادية أخرى مثل الممر والرف الموجود فيه السلعة ويتسبب في أن ترسل السلع للعميل مع معالجته لكل تفاصيل المحاسبة على طول الطريق. ويتسلم العميل السلع التي تم معالجتها عن طريق الرجل الوسيط فقط وكتالوجه الوصفي.



شكل 5.1
قسم الأوامر بالبريد كرجل وسيط

هناك مثل آخر للرجل الوسيط وهو عامل معلومات الهاتف للمسافات الطويلة والذي يعمل كسطح بيني بين الشخص الذي يريد الاتصال برقم هاتف معين وبدلائل الهاتف المختلفة غير المتاحة لهذا الشخص، شكل 5.2. وهناك اختلافات عديدة بين هذا المثال ومثال الشراء بالكتالوج. فالسلعة في هذه الحالة وهي أرقام هاتف في دلائل هي بيانات. وليست قابلة للانتهاء not exhaustible مثل السلعة المشتراه عبر الكتالوج الموجود على أحد الأرفف. ويمكن للعامل أن يحصل على نفس رقم الهاتف أي عدد غير محدد من المرات ويظل الرقم بالرغم من ذلك موجود في مكانه. كما يمكننا أن نعطي كل شخص في البلاد اتصالاً مباشراً لكل دليل هاتف في البلاد بإعطاء كل منهم نسخة من كل دليل من دلائل الهاتف. ودون الدخول في المشاكل الطبيعية المصاحبة لعمل ذلك فسوف نقول ببساطة إن الحل ينتج عنه تكرار هائل للبيانات وليس هناك حاجة لإعطاء كل شخص نسخة من كل دليل من دلائل الهاتف إذا كان عامل المسافات الطويلة

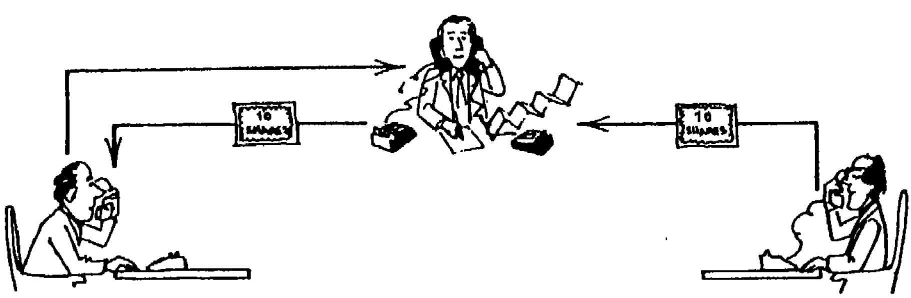
يمكنه العمل كرجل وسيط ويمد العملاء بالأرقام المطلوبة عند حاجتهم لها.



شكل 5.2

عامل الهاتف لاتصالات المسافات الطويلة كرجل وسيط

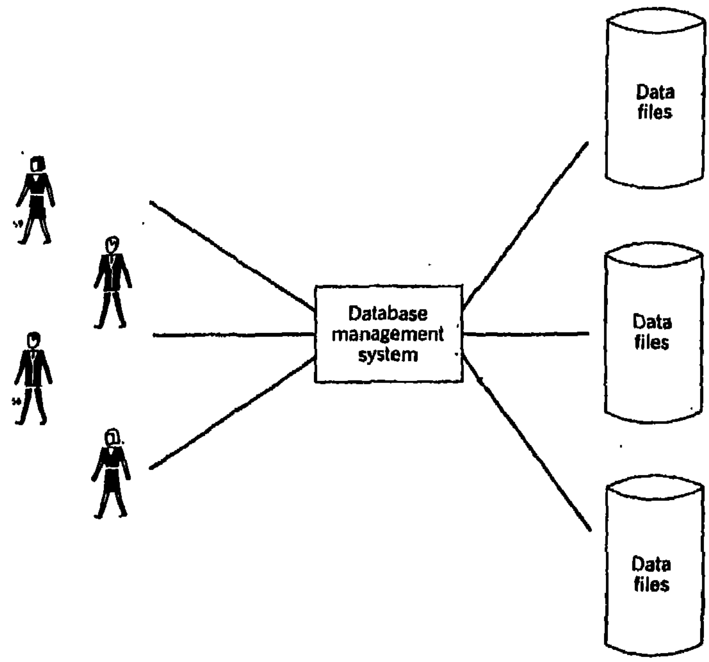
مثال ثالث للرجل الوسيط هو سمسار الأوراق المالية الذي يقوم بتوصيل المشتري بالبائع من خلال سوق الأوراق المالية، شكل 5.3. وهناك اختلاف بين هذا المثال للرجل الوسيط للسطح البيني والمثالين السابقين وهو أنه في هذه الحالة يكون الطرفين متساويين وظيفياً ويمرران بيانات هامة عبر السطح البيني في كل من الاتجاهين. كما أن هذه الحالة هي أوضح حالة من الثلاث حالات في حفظ كل من الطرفين منفصلين عبر السطح البيني للرجل الوسيط. هل يمكنك أن تتخيل مقدار الفوضى من وجود كل من آلاف البائعين والمشترين للأوراق المالية في حجرة واحدة؟



شكل 5.3

سمسار الأوراق المالية كرجل وسيط

نظام إدارة قاعدة البيانات DBMS هو مثل الرجل الوسيط كذلك. فهو يخدم في بيئة تشغيل البيانات والأفراد الذين يسعون الى بيانات في هذه الملفات، شكل 5.4. وسواء كان الاتصال بهذه الملفات يحدث في صورة بحث خاص يكتب في نهاية طرفية تعرضه أو في صورة تقرير رسمي مطبوع فيجب أن يمر الطلب خلال DBMS ويجب أن تمر الاستجابة عبر DBMS كذلك.



شكل 5.4
نظام إدارة قاعدة البيانات كرجل وسيط

ونظام إدارة قاعدة البيانات DBMS هو جزء معقد من نظم البرامج التي تحدد كيفية تكوين البيانات ويراقب كل الاتصالات بالبيانات ويقدم خدمات بيانات معينة ضرورية أخرى. تكوين البيانات في بعض الحالات يكون متماسكاً مع الطريقة التي تعامل بها البيانات ويجب أن يتيح سلامة للبيانات ويقلل من تكرار البيانات ويعبر عن العلاقات المتعددة في البيانات. ويقدم الاتصال بالبيانات عبر تسهيلات استفسارات ومن خلال استدعاءات موجودة في لغات مرتفعة

المستوى أو باستخدام الطريقين. وتقدم نظم DBMS أخرى خدمات تشمل أمن البيانات وعمل احتياطي واستعادة ومراقبة التجديد والتزامن.

وعادة ما تعمل نظم إدارة قاعدة البيانات تحت تحكم غرض عام لنظام التشغيل، عادة ما يأخذ ميزة تسهيلات نظام التشغيل. ويمكن أن يكون لها نظام كمبيوتر خاص بها، أو يمكن أنها تقتسم وحدة تشغيل مركزية مع أنواع تسهيلات نظم البرامج الأخرى. يمكن أن يعمل DBMS والبرامج المصاحبة معه في وسط تشغيل الدفعة أو بيئة خط مفتوح متعددة النهايات الطرفية مع استخدام تسهيلات اتصالات بيانات إضافية أو يمكن أن يعمل في كل من البيئتين في نفس الوقت. كما أن DBMS هو رجل وسيط كذلك. وكما في مثال الشراء بالكتالوج فإن DBMS هو وسيلة الاتصال الوحيدة بالسلع وهي البيانات في الملفات. ومثل مثال معلومات هاتف المسافة البعيدة فإنه يتجنب التكاثر لنسخ نفس البيانات. ومثل مثال سمسار الأوراق المالية فإنه يمرر البيانات في كل من الاتجاهين عبر سطحه البيني ويسمح بقراءة البيانات مع ملفاتها ويسمح بوضع بيانات جديدة فيها أو تعديل بيانات موجودة بالفعل. وكما في كل الأمثلة فهو يقدم درجة أمان للمادة (البيانات) تستجيب للأوامر.

دعنا نلقي نظرة على بعض هذه الأفكار بشيء من التفصيل.

لغات تعريف البيانات: Data Definition Languages

مقدمة:

اصطلاح لغة تعريف (أو وصف) البيانات DDL هو اسم واسع لوجه نظام إدارة قاعدة البيانات الذي يحدد الطريقة التي يمكن أن تخزن بها البيانات. وكما تبدو العبارة أساسية فهناك في الواقع وجهتين نظر للطريقة التي تخزن بها البيانات. بالنسبة للإصدار هناك اختلافان ظاهران بين الأمرين. أحدهما هو الرغبة في أن المبرمج الذي يستخدم البيانات يدرك أنها مخزنة بطريقة بسيطة نسبياً. والأخرى هي الرغبة في وجود كل البيانات كما تخزن فعلاً على وحدات أقراص تتمتع بأقصى مزايا التعقيد في نظم المكونات ونظريات تقنية قاعدة البيانات لتحقيق كل المزايا المعلنة لمنهج قاعدة البيانات. هذا التمييز مهم بدرجة كافية لتحذير مصممي

DBMS بتكاليف معتبرة لانتاج مقدرة وجود وجهتي النظر للبيانات. نحن نتكلم عن وجهة النظر المنطقية logic view للبيانات كصورة يدركها المبرمج بينما وجهة النظر الطبيعية physical view تعكس الطريقة التي تخزن بها بالفعل على القرص.

ولكي يعمل هذا التمييز فيجب أن يمكن لأحد مكونات DBMS تحويل طلبات البيانات التي تأتي من النهايات الطرفية أو من برامج التطبيقات مبنية على وجهة النظر المنطقية الى صورة طبيعية تحقق ذلك. ويجب أن يكون من الممكن عند تنظيم وصفها أن تفهم طريقة تخزين البيانات على القرص وطريقة إدراك المبرمج أو المستفيد لها. مثل هذه المعلومات تخزن في مجموعات سجلات خاصة تسمى مجموعات مراقبة control groups.

الأربعة مناهج للتكوين:

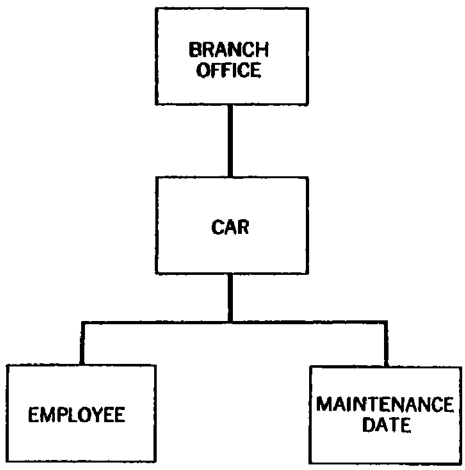
مقدمة: أي DBMS خاص يبني على تكوين البيانات وعلى آلية اتصال بالبيانات. وأياً كان هذا التكوين وهذه الآلية فيجب أن يكونا قادرين على تخزين بيانات بطريقة غير متكررة لمعالجة المصاحبات المتعددة للبيانات ولتكامل البيانات ولتوفير اتصال مباشر وتتابعي للبيانات. وعبر السنوات تم إعداد العديد من نظم DBMS لبيئات تتراوح من أكبر أجهزة الكمبيوتر الكبيرة إلى أبسط أجهزة الميكروكمبيوتر. وكل منها كان مبيناً على أحد الأربعة مناهج الأساسية لتكوين البيانات. وآلية الاتصال بالبيانات الأربعة الخاصة بنظام DBMS وهي:

المنهج الهرمي.
المنهج الشبكي.
المنهج العلاقي.
المنهج شبه العلاقي أو منهج الملف المسطح المتكامل.

توضيح كل من هذه المناهج يحتل جزءاً كبيراً من بقية الكتاب. إلا أن بعض الملاحظات البسيطة مناسبة في هذا المقام.

قواعد البيانات الهرمية: يبين شكل 5.5 هرماً من أنواع السجلات تمثل نوع التكوين الموجود في نظام إدارة قاعدة البيانات DBMS الهرمي. تمثل الأربعة أنواع

سجلات في شكل 5.5 مكاتب فرعية للمؤسسة وسيارات محددة لكل مكتب فرعي وعاملين مسؤولين عن قيادة سيارات محددة وتواريخ صيانة تقدم فيها الخدمة للسيارات. وبالرغم من عدم الوضوح في الشكل فكل من هذه المستطيلات يمثل في الواقع تجميعاً من الحقول كما يجب أن يكون أي سجل. مثال ذلك يحتوي سجل BRANCH OFFICE على حقل BRANCH OFFICE NUMBER وحقل BRANCH OFFICE NAME وغيرها من الحقول. سجلات BRANCH OFFICE و EMPLOYEE تعرف بواسطة حقول رئيسية فردية (مثل BRANCH OFFICE NUMBER لسجل BRANCH OFFICE).



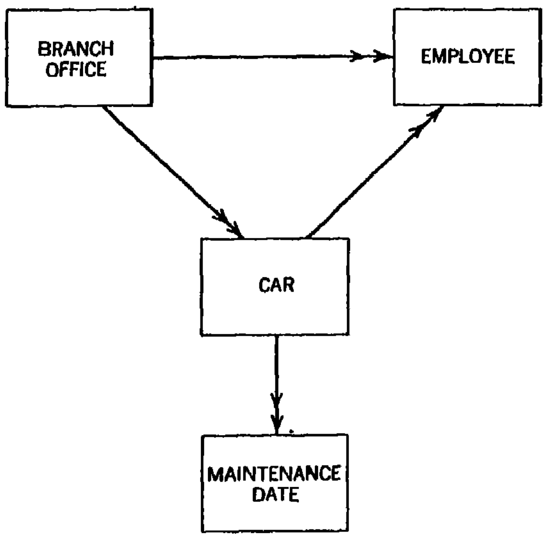
شكل 5.5
هرمية البيانات

الاعتقاد الرئيسي للتكوين الهرمي للبيانات هو أن كل اتصال يصل نوعي سجلات يمثل علاقة واحد للعديد في الاتجاه من أعلى لأسفل. وعلى هذا يحدد تكوين شكل 5.5 إن كل مكتب فرعي له عدة سيارات مخصصة له وكل سيارة لها عدة سائقين محددين لقيادتها وكل سيارة لها تواريخ متعددة للصيانة. ويمكن معاملة علاقة العديد للعديد بخليط من الهرميات كما سيتضح ذلك فيما بعد.

يأتي تقليل التكرار في هذا الترتيب من حقيقة أن تفاصيل سجل أعلى خاص (وليكن سجل BRANCH OFFICE) تخزن مرة واحدة فقط ويمكن أن يتكامل مع تفاصيل أي من حدوث السجلات التابعة (مثل CAR).

وعلى سبيل المثال يمكننا أن نكون قادرين على دمج عنوان مكتب فرعي - المخزن مرة واحدة فقط في حدوث واحد لسجل BRANCH OFFICE لهذا الفرع - بتفاصيل عن أي من السيارات المحددة للفرع. كما يمكن كذلك تحقيق طريقة علاقات العديد للعديد من خلال هرميات متصلة تحفظ عن التكرارات.

قواعد البيانات الشبكية: يبين شكل 5.6 شبكة. وكما في الحالة الهرمية فإن كل اتصال بين نوعي سجلات في الشبكة يمثل علاقة واحد للعديد بينهما.واتجاه العلاقة مبين بواسطة أسهم ذات رؤوس مزدوجة في الشكل. مرة أخرى تتطلب علاقات العديد الى العديد بعض التكوينات الإضافية. وعلى عكس الحالة الهرمية فمن الشائع في الشبكات، كما سوف نرى ذلك بالتفصيل فيما بعد، اعتبار نوعي سجلات متصلان والاتصال الموجود بينهما يكون تكويناً جزئياً يسمى فئة set وله خواص معينة مرتفعة القيمة.



شكل 5.6
شبكة بيانات

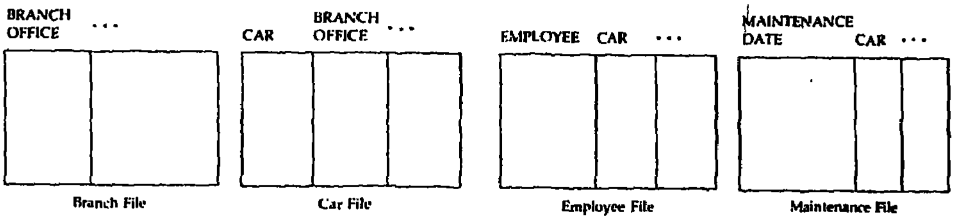
نظم إدارة قواعد البيانات المعتمدة على المنهج الهرمي أو المنهج الشبكي عادة ما تسمى نظم مبحرة navigational. وفي كل من الحالتين تخزن البيانات كسجلات من أنواع مختلفة تكون متصلة بواسطة مشيرات عناوين. والمبرمجون

الـذين يستخدمـون أي من الصيغ الهـرمية أو الشبكيـة يمكنهم أن يبحروا navigate خلال التكوينات. هذا يعني على سبيل المثال أن استدعاء برنـامج لقـاعدة بيـانات يمكن أن يصمم لإيجاد مكتب فرعي محدد وسيارة محددة في هذا الفرع وإنتاج قائمة بالأفراد المسؤولين عن قيادة كل سيارة.

قواعد البيانات العلاقية: يبين شكل 5.7 قاعدة بيانات عـلاقية. ومنـذ أول وهلة تبدو أنها تحتوي على عدة ملفات خطيـة مستقلة. وفي الواقـع هذا صحيـح من وجهـة نظر التكـوين. لاحظ كذلـك أنه حيث أن حقـول BRANCH OFFICE و CAR و EMPLOYEE و MAINTENANCE (بالنسبة لسيـارة معينة) هي حقـول فرديـة فيكون هنـاك سجل واحـد فقط لمكتب فرعي معـين (وهو في ملف الفروع Branch) وسجل واحد لسيارة محـددة (في ملف السيارات Car) وهكـذا. وهذا يعني بالطبع أنه لا يكون هنـاك تكرار عـبر الحقول غـير الرئيسيـة nonkey. وحيث أنه هناك سجل واحد عـلى الأقل لمكتب فـرعي معين فتعتمـد كل الحقـول عليه مثل BRANCH MANAGER و BRANCH ADDRESS وغيرها تظهر مرة واحدة فقط.

لكن مـاذا عن الحقيقـة بـأن حقـل BRANCH OFFICE يـظهـر في ملف السيـارات Car؟ أليس هذا تكـرار مع حقـل BRANCH OFFICE الموجـود في ملفـا الفــروع Branch؟ بمفهـوم معـين نعم ولكن في هـذه الحـالـة يكـون من الضروري حدوث التكرار. حيث أن أنواع سجلات الملفـات في قاعـدة البيانـات العلاقية ليست متداخلة مع بعضهـا في تكوينهـا كما في حـالة منـاهج الإبحـار فإن الطريقة الوحيدة لتكامل الملفات هي تكرار الحقل الرئيسي لأحـد الملفات في ملف آخـر. فإذا لم يكن لـدينا حقـل BRANCH OFFICE في ملف السيـارات Car فكيف يمكن معرفة السيارة المحددة لمكتب الفرع هذا؟ وبالمناسبـة هذه هي طـريقة تنفيذ علاقات واحد الى العديد في قواعد البيانات العلاقية. يمكن أن يظهر العديد من السيارات المختلفة في ملف السيارات على أنها خاصة بنفس المكتب الفـرعي. وسوف نرى فيها بعد أن علاقات العديد للعديد تعامل بنفس الطريقة تقريباً.

وبالرغم من أن الملفات في مفهوم قـاعدة البيـانات العـلاقية مستقلة إلا أنـه يجب أن توجد طريقة لتكامل البيانات لعمل خليط مفيد من البيـانات من ملفـات



شكل 5.7

قاعدة بيانات علاقية

مختلفة. وفي كلمات أخرى يجب أن يكون هناك مكافيء وظيفي لفكرة الإبحار خلال التكوين الهرمي أو الشبكي. وهناك مثل هذا المكافيء إلا أن قوته كما سيتضح فيما بعد تأتي من اجراءات وقت التنفيذ التي تشمل الاتصالات الطبيعية الموجودة في النظم المبحرة.

قواعد البيانات شبه العلاقية: يبين شكل 5.8 قاعدة بيانات شبه علاقية أو قاعدة بيانات من ملفات مسطحة متكاملة. أولاً لاحظ أن الملفات Branch File و Car File و Employee File و Maintenance File متطابقة مع ملفات قاعدة البيانات العلاقية الموجودة في شكل 5.7. والتعليقات السابقة عن تجنب التكرار في الملفات المسطحة في قاعدة البيانات العلاقية تسري هناك كذلك.

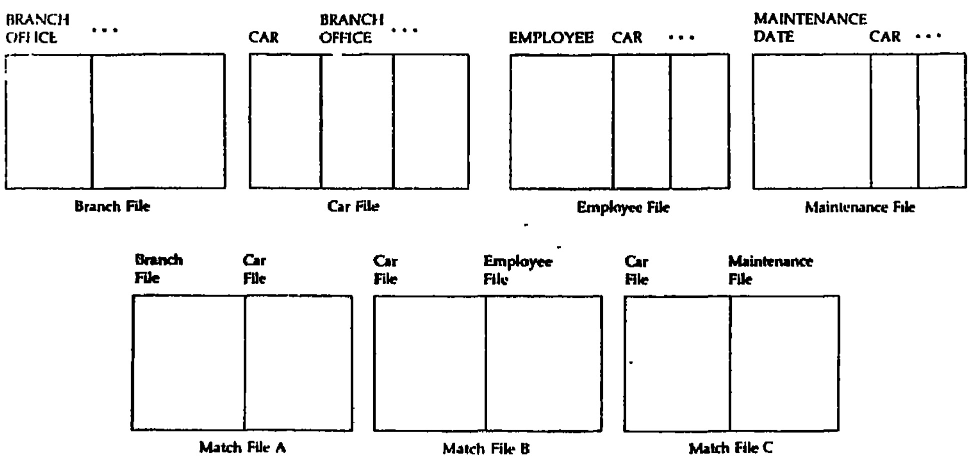
إلا أن إجراءات وقت التنفيذ لتكامل البيانات في الملفات العلاقية لا توجد هنا. ويجب أن يكون هناك طريقة طبيعية مبنية داخلياً لتوصيل الملفات. النصف السفلي من شكل 5.8 يبين ثلاثة ملفات توصيل match إضافية كل منها به حقلين. مثل هذه الملفات تحدد أي سجلات أحد الملفات ترتبط بأي سجلات الملف الآخر ليحدث تكامل كفوء للملفين. مثال ذلك يحدد الملف Match File A أي سجلات في Branch Office File وأي سجلات في Car File مرتبطة ببعضها وهذا يعني بالفعل أي سجلات من هذين الملفين لها نفس أرقام المكاتب الفرعية. يسمح هذا بدمج بيانات تفصيلية عن مكتب فرعي معين مع بيانات تفصيلية عن السيارات المحددة لهذا المكتب. وعلى هذا تدمج قاعدة البيانات شبه العلاقية تكوين بيانات الملف المسطح لقاعدة البيانات العلاقية مع مفهوم concept تحقيق تكامل البيانات بنوع من التخزين المسبق والتكوين الطبيعي كما هو الحال في النظم المبحرة.

لغات معالجة البيانات: Data Manipulation Languages

يشير اصطلاح لغات معالجة البيانات إشارة عريضة إلى الآلية المستخدمة في استرجاع البيانات المخزنة في قاعدة بيانات. وهناك طريقتان أساسيتان للاتصال بالبيانات الموجودة في قاعدة بيانات إحدى هذه الطرق هي وجود برنامج تطبيقات يصدر أمراً لنظام DBMS لإيجاد بيانات معينة في قاعدة البيانات وإعادة الى البرنامج. ونتكلم على هذه الأوامر أو التعليمات على أنها عبارات محتواه «embed-ded statements». الطريقة الأخرى للاتصال بالبيانات هي أن يجلس الشخص إلى نهاية طرفية ويصدر أمراً بلغة خاصة بصورة مباشرة الى نظام DBMS لإيجاد بيانات معينة وإعادتها الى الشاشة ونسمي ذلك بعملية استفسار query. ويمكن استخدام أي من الطريقتين مع الأربعة مناهج لتكوينات قواعد البيانات.

العبارات المحتواة:

منطق برنامج الكمبيوتر سواء كان مكتوباً بالكوبل أو بلغة PL/I أو بلغة المجمع أو بغيرها من اللغات الإجرائية، يذكر أنه عند نقاط تنفيذ مختلفة للبرنامج يجب إحضار بيانات من ملفات مخزنة مناسبة ويجب أن ترسل البيانات الى ملفات مخزنة مناسبة. وعلى هذا فإن الاحتواء embedded في البرنامج في الأماكن المنطقية المناسبة مثل Go To و IF - THEN - ELSE و A × B + C والعبارات



شكل 5.8
قاعدة بيانات شبه علاقية

الأخرى هي عبارات أخرى والتي عندما تنفذ تتسبب في وصول النظام الى مشغلات أقراص (أو مشغلات شرائط وغيرها) ونقل بيانات الى ومن البرنامج الذي ينفذ. وعندما يعمل البرنامج في بيئة قاعدة بيانات فيجب أن تعد عبارات المدخلات والمخرجات بصفة خاصة لتناسب قواعد نظام DBMS والتي تخدم كما سبق ذكره كرجل وسيط بين البرنامج وقاعدة البيانات.

لغات الاستفسار:

هناك مواقف تسمي ببساطة بالبحث عن شيء معين في قاعدة البيانات. في مثل هذه الأوقات يريد الشخص أن يكون قادراً على إرسال أمر مباشر إلى نظام DBMS يفيده بأن يجد البيانات المطلوبة ويرسلها على عكس وجود خطة وكتابة وتصحيح ثم تنفيذ برنامج في النهاية لتحقيق نفس الشيء. مثل هذا التداخل مع DBMS عن طريق نهاية طرفية هو طبيعة لغة الاستفسار query language. منطق تفسير وأداء الاستفسارات يكون معداً على هيئة مجموعة كجزء من نظام DBMS أو كمجموعة قريبة من نظام DBMS. ويجب أن تعد الاستفسارات بصفة عامة طبقاً لتكوين مخطط.

مقارنة العبارات المحتواه مع لغات الاستفسار:

هناك نقطة حاسمة يجب ذكرها في فهم طبيعة لغات معالجة البيانات وليس الوقت مبكراً جداً لعمل ذلك. مفهوم ما يحتويه الاستفسار بالضبط يعتمد جزئياً على وجهة النظر التي نعالج بها الموقف. ففي مواقف لغة الاستفسار هناك مشغل أغراض عامة قادر على البحث عن البيانات واسترجاعها طالما أنها مخزنة طبقاً لتكوين نظام DBMS المفروض وطالما أن تكوين الاستفسار نفسه يتبع الاصطلاحات المناسبة. وأي شخص يستخدم لغة استفسار يجب أن يشعر أنه كما لو كان يجري استفساراً.

دعنا الآن ننظر إلى موقف العبارات المحتواه. أساليب البرمجة الحديثة بصفة عامة ونظم DBMS بصفة خاصة تسمح بكل تأكيد بعمليات موجهة للعمل في وسط الخط المفتوح. ومن المعتاد جداً لتنفيذ برامج بها عبارات مدخلات ومخرجات مكتوبة كاستدعاءات محتواه لنظام DBMS ان تقبل طلبات بيانات من وحدات

النهايات الطرفية وتجدد قاعدة البيانات من قيم البيانات المكتوبة فيها عند هذه الوحدات. والنقطة هي أن المبرمج كان حريصاً عند كتابته البرنامج في المقام الأول على أنه لا يتعامل مع ما نسميه بلغة استفسار ـ ولكن إذا حدث أثناء تنفيذ البرنامج أنك سألت أحد المستفيدين النهائيين end user ما يفعله فيكون رده «استفسار من قاعدة البيانات». أي عملية على بيانات من نهاية طرفية يمكن أن يطلق عليها استفسار ولكن يجب النظر إلى ما يدعم العملية لتحديد ما إذا كانت عبارة محتواه أو لغة استفسار.

وفيما بعد في هذا الكتاب سوف نرى بعض أمثلة لكل من استدعاءات العبارات المحتواه ولغة الاستفسار.

الأمن والاحتياطي والاستعادة والتزامن :

Security Backup and Recovery, and Concurrency

هناك اهتمامات معينة بالنسبة إلى البيانات المخزنة يجب أن ننتبه إليها بغض النظر عن تكوين التخزين المستخدم وبغض النظر عما إذا كان هناك نظام DBMS أم لا. أحد الاهتمامات هو أمن security البيانات. مع ازدياد أعمال المؤسسة التي تنفذ باستخدام الكمبيوتر يتزايد موضوع أمن البيانات بل ويصبح حرجاً. فبالإضافة إلى الحاجة لحفظ كل سجلاتها الداخلية والخاصة بالعملاء دقيقة فهناك أمور لحماية منتجاتها الجديدة وأفكارها الحديثة وحماية سرية بيانات افرادها وعملائها وحماية معلومات ميزانياتها وتخطيطها الاستراتيجي داخل الشركة.وهذه ما هي إلا أسماء فقط لبعض الأمور.

يمكن أن نسأل مثل هذه النظم أسئلة مثل: هل البيانات محمية ضد الاتصال غير المسؤول أو التغييرات غير المسؤولة؟ هل البرامج التي لها اتصال مسموح به بالبيانات محمية ضد التعديلات غير المسؤولة التي يمكن أن تؤثر على البيانات؟ هل الأفراد المسموح لهم بالاتصال ببعض البيانات الموجودة في قاعدة البيانات غير قادرين فعلاً على الاتصال سواء عن طريق الخطأ أو بالتعمد إلى بقية بيانات قاعدة البيانات؟

هناك اهتمام آخر وهو المقدرة على إعادة تكوين البيانات إذا دمر جزء منها عن طريق الخطأ. عملية إعداد نسخة من ملف لمثل هذه الأغراض يشار إليها بأنها عملية احتياطي backup. عملية إصلاح كم بسيط نسبياً من الضرر الذي لحق بملف أو إعادة إنتاج ملف تم تدميره بالكامل تسمى بعملية استعادة recovery.

هناك اهتمام ثالث وهو موضوع التزامن concurrency. كما سيتم تطويره فيما بعد يمكن أن تظهر مشاكل عندما يحاول مستفيدان أو أكثر تجديد جزء معين من البيانات في نفس الوقت متزامنين. فبدون حماية قد لا تأخذ النتيجة النهائية في الحسبان على الإطلاق بالنسبة للتغييرات المطلوبة.

وحيث أن كل هذه الاهتمامات الثلاثة هي اهتمامات مشتركة لكل البيانات. وحيث أن الإجراءات اللازمة لمعاملتها يمكن أن تعد بطريقة عامة جداً فمن الواضح أن مثل هذه الاجراءات يجب أن تعد داخل أي نظام DBMS. وعمل شيء آخر غير ذلك يعيدنا إلى الأيام الأولى لتشغيل البيانات عندما كان المبرمجون يفعلون ما يريدون بالنسبة لهذه الأمور مع كل تطبيق جديد يعدونه.

السطح البيني لاتصالات البيانات

Data Communications Interface

لقد ناقشنا في جزء من هذا الفصل الاستفسارات وعمليات الاستفسار وأخذنا أي تفاصيل ضرورية على أنها موجودة وقد افترضنا أن التداخل مع قاعدة البيانات كان ممكناً من خلال نهاية طرفية للعرض المرئي.

ولكن في الواقع كل المادة التفصيلية التي نوقشت حتى الآن في هذا الكتاب ومعظم ما سيأتي فيما بعد خاص بالبيانات ونظم DBMS ومواضيع مرتبطة ارتباطاً مباشراً بالاتصال بالبيانات. وكل هذا يمكن أن يحدث في وسط التشغيل بالدفعة. إلا أن معظم أوساط DBMS الحالية تعمل بنظام الخط المفتوح أساساً وذلك لتحقيق اتصال سريع بالبيانات وإجراء تغييرات فورية على البيانات وعمل اتصال بالبيانات في نطاق جغرافي واسع. ويمكن أن يوجد المستفيدون ونهاياتهم الطرفية على هيئة مجموعات في نفس الموقع بالقرب من الكمبيوتر أو يمكن أن يكونوا

بعيدين جداً عبر البلاد أو في جميع أنحاء العالم.

الآليات اللازمة للسماح بمثل هذه المنظومة من النهايات الطرفية وبصفة عامة بمحتويات بيئة الخط المفتوح ليست تافهة ولا تكون حرة. وسواء كانت البيانات في قاعدة بيانات أم لا فإن الاتصال بها في وسط الخط المفتوح يحتاج إلى جزء معقد من نظم البرامج عادة ما يسمى بمشغل اتصالات البيانات Data Communica-tion Driver وذلك مع الأسطح البينية مع نظام DBMS وبرامج التطبيقات (لاحظ أن التعليق على وجود قاعدة بيانات أم لا هو بنفس المعنى مثل التعليق الشبيه الخاص بالأمن وخلافه في القسم الأخير).

وفي صورة الاتصال بلغة الاستفسار يجب أن يكون نظام DBMS معداً لاستقبال رسائل النهايات الطرفية وإرسال رسائل لها. وفي صورة العبارات المحتواة يجب ألا تحتوي برامج التطبيقات على إمكانية استخدام تعليمات خاصة للاتصال بقاعدة البيانات فقط بل يجب أيضاً أن يكون لديها المقدرة على استخدام تعليمات خاصة أخرى للتداخل مع النهايات الطرفية. هذه التعليمات التي يتم ادخالها في مواقع مناسبة من الشفره طبقاً لمنطق البرنامج يجب أن تكون قادرة على قبول رسائل من النهايات الطرفية من خلال مشغل اتصالات البيانات وترسل رسائل للخلف مرة أخرى. والرسائل التالية في معظم حالاتها الأساسية سوف تطلب لبيانات موجودة في قاعدة البيانات وتعليمات لتغيير بيانات موجودة في قاعدة البيانات والرسائل الخارجة هي بيانات ترسل للخارج عبر النهايات الطرفية التي تطلبها.

وسوف ننظر إلى مواضيع الأمن والاحتياطي والاستعادة والتزامن بتفاصيل أكثر فيها بعد في هذا الكتاب.

مراجع: References

Date, C. J., *An Introduction to Database Systems*, 3rd ed., vol. 1, Addison-Wesley, Reading, MA, 1981.

King, J. M., *Evaluating Data Base Management Systems*, Van Nostrand Reinhold, New York, 1981.

Kroenke, D., *Database Processing*, 2nd ed., Science Research Associates, Chicago, IL, 1983.

Martin, J., *Computer Data-Base Organization*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1977.

Ross, R. G., *Data Base Systems*, AMACOM, New York, 1978.

اسئلة وتمارين: Questions and Exercises

١ ـ كما سبق وصفه في هذا الفصل فإن نظام إدارة قاعدة البيانات يلعب دور الرجل الوسيط بين البيانات والمستفيدين. فكر في مثال آخر لدور الرجل الوسيط خارج مجال تشغيل البيانات. هل يقدم ذلك فصلاً كاملاً بين العنصرين؟ هل يعتبر العنصران الرجل الوسيط أنه مساعد أو أنه عائق؟

٢ ـ ماذا تعتقد أن تكون مزايا وجود رجل وسيط بين البيانات والمستفيدين منها في بيئة تشغيل البيانات؟ هل هناك أي عيوب؟

٣ ـ بكلمات من عندك حدد ما نظام إدارة قاعدة البيانات.

٤ ـ اعمل تخطيطاً للإمكانيات التي يتوقع أن توجد لدى نظام ادارة قاعدة البيانات.

٥ ـ ما الاختلاف بين وجهتي النظر المنطقية والطبيعية للبيانات المخزنة في قاعدة بيانات؟ ما أسباب عمل هذا التمييز؟

٦ ـ ما الاختلاف بين لغة تعريف البيانات ولغة معالجة البيانات؟ وما درجة ارتباط كل منهما بالأخرى؟

٧ ـ ما الأربعة مناهج لتكوين نظام ادارة قاعدة البيانات؟ من النظرة الأولى هل تستطيع أن تفكر في بعض منها على أنها أفضل من الأخرى؟ لماذا ولم لا؟

٨ ـ من النظرة الأولى هل تعتقد أن كل الأربعة مناهج لنظام إدارة قاعدة البيانات قادرة على إشباع كل الاحتياجات من قاعدة البيانات؟

٩ ـ ما الاختلاف بين لغة معالجة البيانات المحتواة ولغة معالجة البيانات من نوع الاستفسار؟

١٠ ـ في كلمات من عندك صف الظروف التي يمكن تحتها لأحد الأشخاص أن يدرك تطبيقاً يستخدم عبارات محتواة ليكون في عالم لغة من نوع الاستفسار.

١١ ـ اكتب وصفاً للأمن والاحتياطي والاستعادة والتزامن. ما درجة أهمية معالجة نظام إدارة قاعدة البيانات لهذه الأمور في رأيك؟

الفصل السادس
المنهج الهرمي لقاعدة البيانات
The Hierarchial Approach to Database

مقدمة Introduction

أول منهج ننظر إليه لنظم إدارة قواعد البيانات DBMS هو المنهج الهرمي. وحيث أن أحد أقدم وأكثر نظم ادارة قواعد البيانات DBMS المستخدمة مبنى على هذا المنهج فسوف يفحص هذا الفصل نظام إدارة المعلومات Information Management System (IMS) لشركة IBM وهو المقصود بأقدم وأكثر هذه النظم استخداماً.

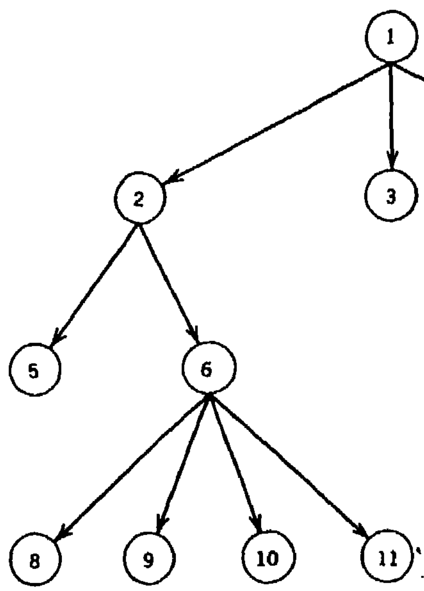
في واقع الأمر يتكون نظام إدارة المعلومات IMS من جزئين جزء منها هو لغة بيانات Data Language/ I (DL/I) وهو وسيلة قاعدة البيانات نفسها ويشمل قواعد تكوين البيانات والسطح البيني للغة البرمجة. والجزء الثاني يسمى وسيلة اتصالات البيانات Data Communications Facility وهو مشغل اتصالات (نظم البرامج التي تعرف بيئة تدخل) ويسمح بإرسال رسائل وطلبات لبيانات من النهايات الطرفية الى الكمبيوتر المركزي وعودة الإجابات الى النهايات الطرفية. ونظام IMS ليس نظام تشغيل في حد ذاته ولكنه يعمل تحت تحكم عائلة نظم التشغيل كبيرة الحجم OS/MVS من شركة IBM ويمكن أن يستخدم DL/I كذلك نظام مراقبة معلومات العملاء Customer Information Control System (CICS) كمشغل اتصالات في أي من بيئتي التشغيل OS أو DOS وهو نظام تشغيل متوسط (الحجم) كما يمكن تشغيله كنظام دفعه فقط بالرغم من أن هذا ليس شائعاً حالياً. وعملياً يستخدم IMS و DL/I كمرادفين بالتبادل للإشارة إلى جزء قاعدة البيانات من النظام وهذا هو الجزء الذي سيناقش في هذا الفصل.

نظام IMS هو نظام إدارة قاعدة بيانات كامل الحجم قادر على معاملة

ملفات كبيرة جداً وعدد كبير جداً من المستفيدين في نفس الوقت. وبالطبع فإنه لديه المقدرة على تكامل البيانات وتقليل التكرار ومعاملة العلاقات المتعددة في البيانات. كما أنه يشمل كذلك تسهيلات للاحتياطي والاستعادة والأمن وحماية التشغيل المتزامن.

الهرميات Hierarchies

قبل النظر إلى نظام IMS يجب أن نضع قاعدة ثابتة لما تعنيه الهرمية hierarchy أو الشجرة tree (ستستخدم الكلمتان بالتبادل). يبين شكل 6.1 شجرة أو هرمية موجزة. وهي ببساطة عبارة عن ترتيب من عقد nodes (مرقمة من 1 إلى 11 في الشكل) متصلة بواسطة أفرع branches أو حافات edges (الخطوط المستقيمة في الشكل) وعليها القيود التالية. وهناك عقدة خاصة تسمى بالجذر root ودائماً ما تكون في قمة الرسم. العقدة 1 هي عقدة الجذر في شجرة شكل 6.1 وتتفرع الشجرة من الجذر لأسفل وكل عقدة قادرة على التفريع إلى عدة



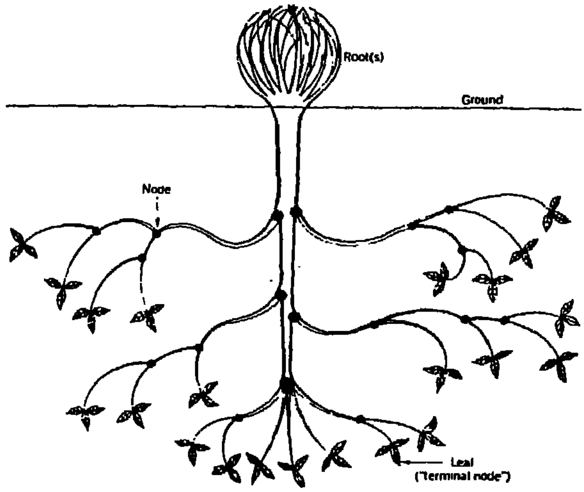
شكل 6.1 شجرة أو جذر

عقد في المستوى السفلي التالي لمستواها. ونتيجة لذلك فكل عقدة (باستثناء الجذر). تكون متصلة لأعلى بالعقدة 1 فقط. العقدة 9 متصلة لأعلى (فوراً) بالعقدة 6 فقط.

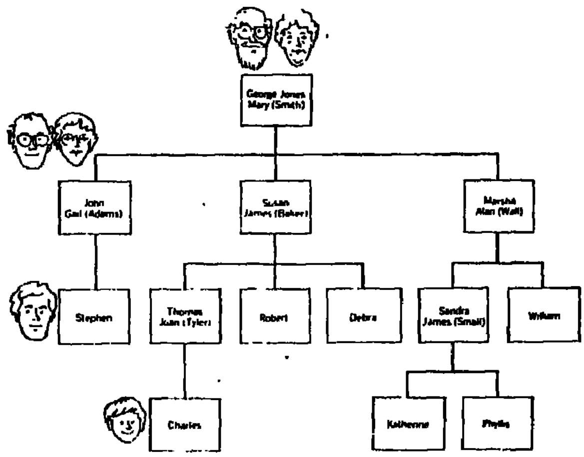
العقد التي تأتي أسفل عقدة معينة مباشرة تسمى بالعقد التابعة child nodes. وكل من هذه العقد التابعة متصلة بعقدة تقع في المستوى الأعلى منها وتعرف بالغقدة العائلة parent nodes. مثال ذلك العقد 2 و3 و 4 في شكل 6.1 هي عقدة تابعة للعقدة 1 والعقد 5 و 6 عقد تابعة للعقد 2. والعقدة 1 هي عقدة عائلة للعقد 2 و 3 و 4والعقدة 2 هي عقدة عائلة للعقد 5 و 6. ويمكننا أن نتكلم عن كل الغقد التي تقع تحت عقدة معينة على أنها سلالات descendants. وعلى هذا فإن سلالات العقدة 1 هي العقد من 2 إلى 11. وسلالات العقدة 2 هي العقد 5 و 6 و 8 و 9 و10 و 11. وتسمى العقدة التي لا يخرج منها فروع (مثل العقد 3 و 5 و 8) بورقة leaf أو بعقدة نهاية terminal node كتمييز لها عن العقد 1 و 2 و4 و 6 والتي تسمى بأنها ليست نقاط نهاية nonterminal nodes أو أنها عقد وسيطة intermediate nodes. والعقد التي لها نفس العائل (مثل العقد 5 و 6) تسمى توائم Twins أو أخوات Siblings.

وفي الواقع فإن استخدام كلمة شجرة هنا والمصطلحات المصاحبة مستخلصة مباشرة من الأشجار الموجودة في حياتنا اليومية. ويبين شكل 6.2 صورة لشجرة مقلوبة من أعلى لأسفل. في القمة يوجد جذرها وتتفرع لأسفل للعقد والأوراق. ويبين شكل 6.3 شجرة عائلة بكل من مفاهيم الأطفال والسلالات والآباء وما إلى ذلك. فكل من George و Mary لهم ثلاثة أطفال هم John و Susan و Marsha وهكذا.

وقد يكون استخدام مفهوم الشجرة في إدارة بيانات مخزنة غامضاً عند هذه النقطة إلا أننا سوف نطور بالتدريج الفكرة في الصفحات التالية.



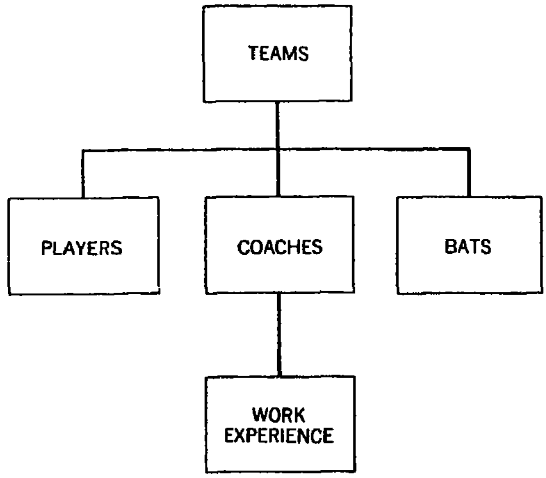
شكل 6.2
شجرة حقيقية مقلوبة



شكل 6.3
شجرة أسرة Johes

التكوين الطبيعي لنظام IMS — IMS Physical Structure

والآن وبعد أن عرفت معنى الهرمية فلنبدأ مناقشة نظام IMS بوصف كيفية إمكانية تخزين البيانات في هذه الصورة وما هي مميزات عمل ذلك. سوف نستخدم مثالاً يشمل فريقاً للبيسبول. وعلينا أن نحفظ معلومات عن كل فريق من الفرق بما في ذلك معلومات عن كل اللاعبين الموجودين في الفريق وكل مدربي الفريق وكل أجزاء العمل السابق الذي يكسب كل مدرب خبرته وكل المضارب (كل مضرب له رقم مسلسل فريد خاص به) لكل فريق.



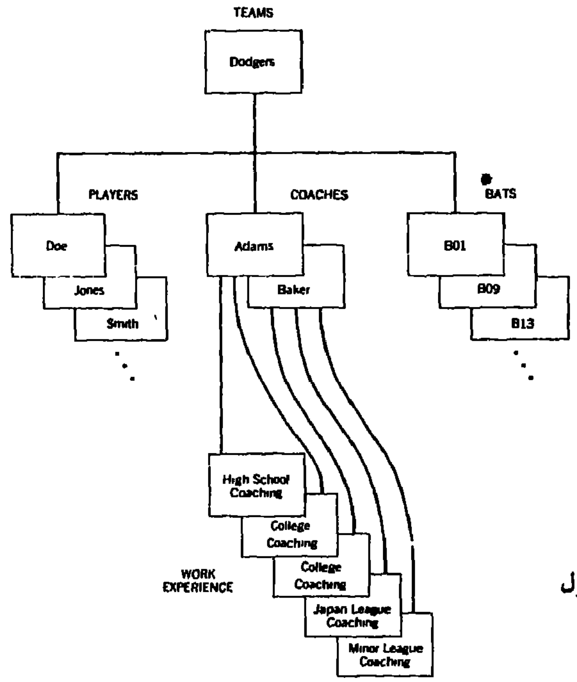
شكل 6.4
هرمية فرق البيسبول

هرميات بسيطة:

هـرمية فـرق البيسبول: يبـين شكل 6.4 هـرمية فـرق البيسبول. أول شيء يمكن التحقق منه هو أن كـل عقدة في هـذه الشجرة تمثـل نوع سجـل record type مختلف. تسمى أجـزاء segments في نظام IMS سجـل أو جزء TEAMS يحتـوي، مثـل أي سجـل، عـلى عـدد من الحقـول التي تصف الفـريق: TEAM NAME و TEAM CITY و TEAM MANAGER ومـا إلى ذلـك. الكلمات المـوجـودة في المستطيلات في شكل 6.4 هي أسماء سجلات أو أجزاء وأسماء الحقول غير مبينة عند هذه النقطة وذلك لجعل الشكل بسيط نسبياً.

يمثل كل نوع في الشجرة الموجودة في شكل 6.4 علاقة واحد للعديد. وعـلى هذا فلفريق معين يوجد هناك العديد من اللاعبين والعديد من المـدربين والعـديد من المضـارب. ولمدرب معـين يوجـد العديـد من أجزاء العمـل التي تكون خـبرته. (وبـالمناسبـة فالعـديد في هـذا المضمون تعني في الـواقع 0 أو 1 أو أكـثر من 1. وعلى سبيل المثال قد لا يكون لدى المدرب أجـزاء عمل تـطبيقية تكـون خبرتـه أو يكون لديه جزء واحد أو أكثر من جزء واحد).

وما يزال يبين شكل 6.4 التخطيط الشامل لكيفية تخزيننا المعلومات الخاصـة بفرق البيسبول. ولا يبين تفاصيل أشكال المخازن الخاصة بكل فـريق من الفرق. ويبين شكل 6.5 حـدوث occurence هرميـة الفريق Team لـ Dodgers. مثـل



شكل 6.5
حدوث Dodgers لهرمية فرق البيسبول

هذا الحدوث الهرمي الذي يسمى سجل قاعدة البيانات data base record في نظام IMS يحتوي على حدوث فردي لعقدة جذر وكل الحدوثات التابعة للأجزاء.

كل سجلات قاعدة البيانات التي لها نفس الهرمية الموضوعة مع بعضها تسمى بقاعدة بيانات database. وبالاستمرار خطوة أخرى للأمام مع هذا التعريف فالعديد من قواعد البيانات التي تتكلم مع بعضها عادة ما تسمى بطريقة غير رسمية بقاعدة بيانات أو ببنك بيانات.

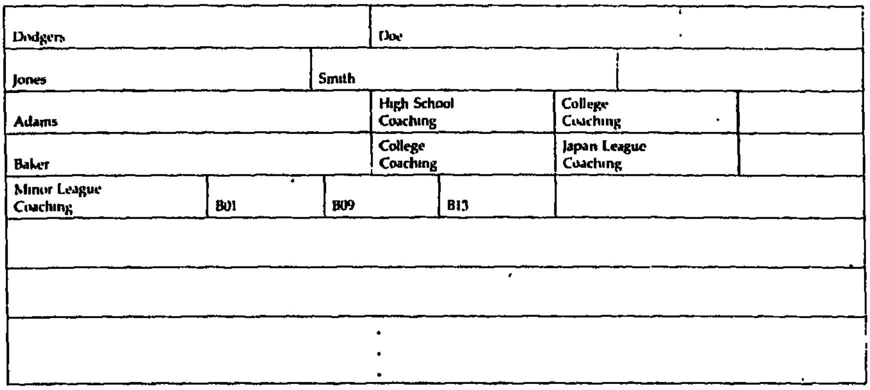
وعلى هذا نجد في شكل 6.5 حدوث Dodgers لسجل TEAM في القمة وحدوث Doe و James و Smith لسجل PLAYERS يصاحبه من أسفل وهكذا. المدرب Adams لديه خبرة تدريب من الكلية والمدرسة العليا. المدرب Baker لديه خبرة تدريب من الكلية ومن اليابان (تهدف خطوط الاتصال الموجودة في شكل 6.5 الى توضيح علاقات واحد إلى العديد وليست ترتيبات التخزين الطبيعي أو الواقعي - التي تأتي فيها بعد). باتباع الملاحظات السابقة الكلمات الموجودة في مستطيلات في شكل 6.5 هي أسماء سجلات أو أسماء حدوث أجزاء (أو قيم حقول رئيسية - يمكنك أن تنظر لها بأي طريقة). وبالرغم من عدم الوضوح إلا أننا نفهم أن سجل Doe يشمل بالفعل حقولاً تصف رقمه المتسلسل

كلاعب واسمه والمدينة التي يقطن بها وما إلى ذلك.

يلاحظ شيء آخر عن ترتيب الهرمية هذا والـذي قد لا يكـون واضحاً عـلى الفور وهو الوفر فيها يمكن أن يكون تكـرار بيانـات. فأيـاً كانت المعلومـات الموجـودة عن Dodgers المخزن في جزء Dodgers مثـلاً فإن اسم مـدير الفـريق يسري على كل الأجزاء التابعة. مثال ذلك في وقت معين أو في وقت آخر قد نريد معرفة مدير Doe أو مـدير Jones أو مـدير Smith وحيث أنهم جميعـاً في نفس الفـريق فكلهم لهم نفس المـدير وفي الـواقع ليس علينـا أن نخزن اسمـه مكرراً مـع كـل لاعب. تحفظ المعلومات مرة واحدة في جزء Dodgers. جزء من نفس تكامل البيانـات في ترتيب الهرمية هو أنه داخل سجل قاعدة البيانات تسري البيانات الموجودة في جزء علوي على كل أحداث الأجزاء التابعة.

مـرة أخرى يحضرنـا هـذا إلى أحـد الأمـاكن حيث يجب أن يعـطي المنـطق الـطريق إلى الـواقعيـة أو الى الـطبيعـة. وتبـدو المنـاقشـة السـابقـة أنها جيـدة ولكن كيف يمكنك إقناع الكمبيوتر أن هناك ما يفيـد حدوث السجـلات من أنواع مختلفة مرتبة ترتيباً يشبه الشجرة والذي سبق أن ذكـرناه؟ هـذا ليس واضحاً وليس تافهاً لكنه مهم جداً وفيما يلي طريقتين اساسيتين.

التخزين التتابعي للهرميات: إحدى طرق تخـزين حدوثـات هرميـات نظام IMS هي نوع من جعلها مسطحة بطريقة محكمة وتخزينها تتابعياً. يبين شكـل 6.6 مخططاً لقسم من ذاكرة الكمبيوتر (ويمكن أن تكون ذاكرة ابتدائية أو ثانوية) تحتوي على سجل قاعدة البيانات Dodgers من شكـل 6.5. حدوثـات الأجزاء وضعت في الخارج في تسلسل معين وهو بديل لـلأسلوب المعروف بـأنه اعـتراض تتابـع سبق تـرتيبه preoder sequential traversal وهـو بالضرورة أسلوب من أعـلى لأسفـل ومن اليسار لليمين ومن الأمام للخلف. لاحظ أننا بدأنا عند قمة الشجرة وأخذنـا جزء Dodgers أولاً. بعد ذلك تحركنا لأسفل ولليسار لالتقاط جـزء اللاعب Doe وحيث أنه لا يوجد شيء تحت Doe فإننا ننتقل إلى الخلف خلال Jones و Smith ثم ننتقل نحو اليمين لنصل إلى المدرب Adams وحدوثات أجزاء تـابعة وهكـذا. هل يمكننا إعادة بناء الشجرة من الصورة المخـزنة في شكـل 6.6؟ نعم ولكن يجب أن يكون لدينا بعض المعلومات الأخرى على طول الطريق مثـل نوع الجـزء وطول الجزء لكل حدوث مخزن. (بالمناسبة الأماكن الفـارغة في نهايـة كل سـطر في شكل



شكل 6.6:

تخزين تتابعي لحدوث فريق Dodgers

6.6 ليست شقيقة مباشرة لهذه المناقشة إلا أنها تضيف اليها بعض الواقعية. فهي تحدد أنه يمكن أن يوجد بغض حدود للأماكن وسط تخزين القرص والسجلات التي تناسب التخزين فقط هي التي تفعل ذلك. فإذا لم يحدث ذلك فيجب أن توضع في سطر تال).

يبدو ذلك أنه يشبه ترتيباً عاملاً لتخزين سجل قاعدة البيانات وفي الواقع فهو مناسب جداً لأنواع معينة من التطبيقات. ولسوء الحظ فهذا له عيبين يجعلانه غير مناسب للاستخدام في معظم تطبيقات قواعد البيانات وبصفة خاصة التطبيقات التي تتطلب استجابات سريعة في وسط خط مفتوح على الاستفسارات. أحد المشاكل هي طريقة وضع حدوثات الأجزاء في الخارج متسلسلة بدقة تجعل من الصعب إدخال بيانات جديدة في منتصف سجل قاعدة البيانات. مثال ذلك إذا التحق لاعب جديد بالفريق فلا يكون هناك مكان لإدخال حدوث جزءه ضمن اللاعبين الموجودين بالفعل في الفريق.

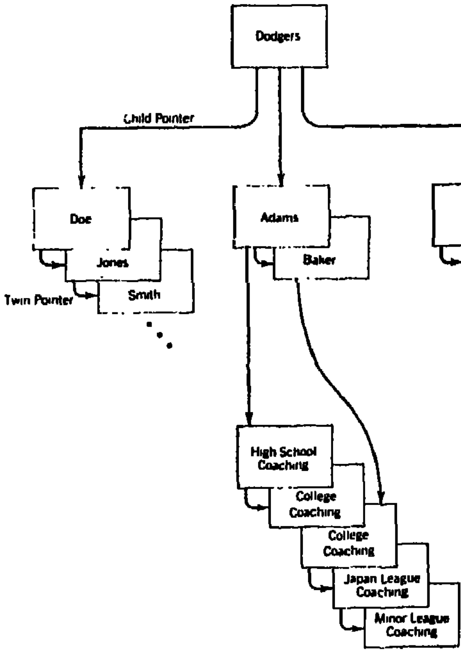
المشكلة الأخرى خاصة بالاتصال السريع بحدوث جزء محدد. دعنا نفترض حالياً أننا نستطيع الوصول إلى كل حدوثات أجزاء الجذور لأجد أنواع الأسس المباشرة (سوف نطور هذا الأمر أكثر فيها بعد). بترتيب التخزين التتابعي لا توجد طريقة للوصول لأي حدوث جزء تابع (أي كل حدوثات الأجزاء غير الجذرية في سجل قاعدة البيانات) فيما عدى الحركة البطيئة تتابعياً خلال الأجزاء المخزنة بدءاً بالجذر وحتى يوجد الجزء المطلوب. مثال ذلك في شكل 6.6 إذا كنت تعرف أن

Baker هو أحد المدربين في Dodgers وتريد أن تعرف خبرته فعليك أن تقرأ خلال جزء Dodgers و Doe و Jones وما إلى ذلك حتى تصل أخيراً إلى Baker وأجزاء خبرته. وهناك في الواقع طريقة أفضل.

تخزين الهرميات المبني على المشيرات: الطريقة الأفضل لتحقيق اتصال سريع بحدوث معين لجزء مبنية على استخدام مشيرات لتصل حدوثات الأجزاء المختلفة لسجل قاعدة البيانات مع بعضها. يبين شكل 6.7 سجل قاعدة بيانات Dodgers المخزن واقعياً مع نوعين من المشيرات معروفان على أنهما مشيرات تابعة child pointers ومشيرات توائم twin pointers. تتبع هذه الاصطلاحات استخدام كلمات تابع child وتوأم twin التي سبق ذكرها. المشير التابع (يسمى بمشير أول تابع واقعي في نظام IMS) يشير من عقدة معينة الى أول حدوث لأحد أنواع أجزائه التابعة. وعلى هذا ففي شكل 6.7 يوجد مشير تابع من Dondgers الى Doe وهو أول لاعبيه وإلى Adams وهو أول مدربيه وإلى Bol وهو أول المضارب. كما يوجد كذلك مشير تابع من Adams الى High School Coaching ومن Baker إلى College Coaching في كل حالة يمثل هذا أول جزء من خبرتهم.

وكل تجميع لحدوثات أجزاء من نفس النوع ينتمي الى نفس حدوث نوع الجزء العائل تتصل مع بعضها بواسطة مشيرات توائم twins (مشيرات «twin واقعية للأمام» في نظام IMS) فكل من Doe و Jones و Smith لاعبين في Dodgers وعلى هذا فهم متصلون مع بعضهم بواسطة مشيرات توائم (والترتيب يسمى بسلسلة توائم) ويتصل المدربان Adams و Baker بسلسلة مع بعضهما تحت Dodgers ويحدث نفس الشيء بالنسبة الى كل المضارب الخاصة ب Dodgers. وبالمثل جزء High School Coaching وأول جزء من College Coaching متصلين مع بعضهما بسلسلة تحت المدرب Adam. كما أن ثاني جزء من College Coaching وجزء Japan League Coaching وجزء Minor League Coaching فهي تحت المدرب Baker.

يمكن إضافة حدوث جزء جديد واقعياً في أي مكان على القرص طالما أن المشيرات المناسبة مجددة وتضاف لحفظ التركيب المنطقي المناسب لسجل قاعدة البيانات. وهذا يساعد لأسباب معينة في الأداء المبني على تقليل كمية الانتقال لآلية ذراع اتصال القرص لإدخال جزء جديد في مكان مفتوح قريباً من عناصر



شكل 6.7
تخزين مشير تابع وتوأم لحدوث فريق Dodgers

أخرى في السلسلة التوأم بقدر الإمكان.

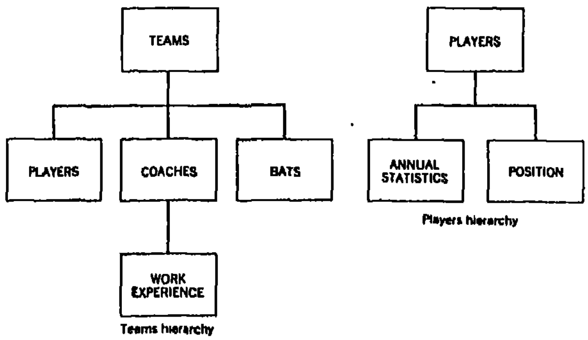
وهذا الترتيب أفضل كثيراً للاتصال الأسرع للأجزاء التابعة مرة أخرى بافتراض أننا يمكننا الحصول على جزء Dodgers مباشرة وتوقيع قطع خبرة Baker يتم تحقيقه في الواقع. أولاً نتتبع مشير تابع من Dodgers إلى Adams (يمكن للنظام أن يميز مشير تابع من مشير تابع آخر وعلى هذا فبالأمر المناسب يصل النظام إلى Adams وليس إلى Doe أو Bol). ثم نتبع مشير توأم من Adams إلى Baker ومشير تابع من Baker الى أول قطعة من القطع الخاصة بخبرته وسلسلة من المشيرات التوائم إلى بقية قطع خبرته.

وكبدائل لأسباب الأداء يقدم نظام IMS كذلك مشيرات تذهب من حدوث جزء إلى آخر حدوث جزء في سلسلة توأم للتوابع.

العلاقات المنطقية:

مقدمة: لقد رأينا أن التكوين الهرمي البسيط يمكن أن يكون صورة تخزين قوية جداً. إلا أنه يمكن أن يكون محدوداً في كل من درجة تكامل البيانات التي يسمح بها وفي الواقع في أنه لا يستطيع معاملة علاقات العديد للعديد مباشرة.

يبين شكل 6.8 تكوينين هرميين. التكوين الموجود على اليسار هو هرمية الفرق Teams الذي سبق أن نظرنا إليه أما الآخر الموجود على اليمين فهو هرمية أخرى وهي هرمية اللاعبين Players. وهرمية اللاعبين لها سجل قاعدة بيانات واحد (تذكر أن هذا يعني حدوث جذر واحد وحدوثات لكل توابعه) لكل لاعب في الفريق. ويخزن مع كل لاعب احصائياته السنوية مرتبة كحدوث جزء واحد في السنة ومعلومات عن موقفه الذي يؤهله للعب.

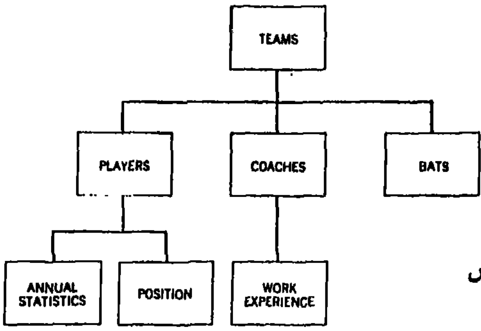


شكل 6.8
هرميات الفرق واللاعبين

ومع نظرك إلى هاتين الهرميتين جنباً إلى جنب فقد يبدو لك غريباً أن جزء PLAYERSيظهر في كل منهما. وهو يمثل على اليسار اللاعبين في إحدى الفرق إلا أنه يمثل على اليمين قائمة مستقلة لكل اللاعبين. دعنا نفترض أن الحقول في كل من جزئي PLAYERS متكافئة (PLAYER SERIAL NUMBER و PLAYER NAME و PLAYER HOME ADDRESS. . . إلخ)، وعلى هذا يجب أن يكون هناك هرمية واحدة بديلة لهاتين الهرميتين في شكل 6.9 مع الاحصائيات السنوية للاعبين ومعلومات مواقفهم موجودة تحت جزء PLAYERS في هرمية الفرق Teams. هل هذا حل أفضل؟

هناك مزايا لكل ترتيب من ترتيبات تخزين البيانات الموجودة في شكل 6.8 وشكل 6.9 (بالنسبة للمناقشة التالية تذكر أن الغرض الذي سبق وضعه في هذا الفصل وهو الاتصال المباشر يمكن أن يحدث على جذر سجل قاعدة البيانات فقط ما زال قائماً).

هناك ميزتان للترتيبين الهرميين الموجودين في شكل 6.8 فقد يكون من الضروري لبعض التطبيقات أن تكون قادرة على الوصول إلى حدوثات أجزاء PLAYERS مباشرة. وفي الترتيبين الهرميين لشكل 6.8 يمكن الوصول إلى



شكل 6.9
هرميات اللاعبين والفرق مدمجه مع بعضها البعض

PLAYERS مباشرة حيث أن جزء PLAYERS هـو جذر لقـاعدة بيـاناتـه. وفي الهرمية الفردية المـوجودة في شكـل 6.9 لا يمكن الوصـول الى PLAYERS مباشرة (بالرغم من أنه فيما بعد سوف نرى زخرفة للنظام تسمح بذلك). كما أنه في شكل 6.8 فإن أحد اللاعبين المنتمي لإحدى الفرق حالياً يمكنه أن يظل موجوداً في هرمية Players المستقلة عن الفرق. وفي الهرمية الفرديـة الموجـودة في شكل 6.9 لا يمكن أن يوجد اللاعب إلا إذا كان منتمياً إلى إحدى الفـرق. (يمكن علاج ذلـك بإيجـاد فريق خاص اسمه «لاعبين غير منتمين لفرق». ولكن يمكنك أن ترى بصفة عـامة أن هذا النوع من الأشياء يمكن أن يكون مشكلة).

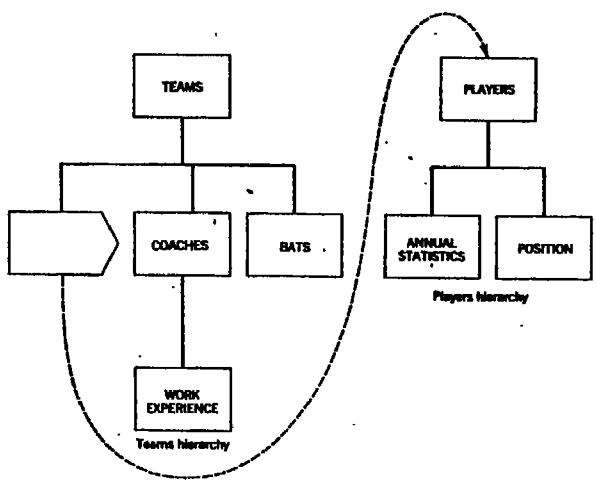
ومن ناحية أخـرى هناك ميـزتان لـترتيب الهرميـة الفرديـة الموجـود في شكل 6.9. الميـزة الأكثر وضـوحاً هي أننـا قد المحنـا بالفعـل الى: بنقـل الاحصـائيـات السنـوية لـلاعب ومعلومات المـوقف الخاصـة به إلى هـرمية الفـرق Teams فـإننـا نحذف التكرار الذي يظهـر جزء PLAYERS مـرتين مـرة في كل من الهـرميتين. وبالترتيب الفردي للهرمية يمكننا أن نحقق درجة سلامة أعلى للبيانات. افرض أننا نريد تقريراً بتفاصيل كل معلومات مواقف اللاعبـين لكل الـلاعبين المـوجودين في فريق معين. بالرغم من أننا لم ندخل في الاتصال بالبيانـات في IMS حتى الآن إلا أنك تستطيع أن ترى أنه في الترتيبـين الهرميـين الموجـودين في شكل 6.8 علينـا أن نستخـدم هرميـة الفرق Team لإيجـاد من هم كل الـلاعبين المـوجـودين في هـذا الفـريق. ثم استخـدام هـذه المعلومـات في البحث عن معلومـات مـواقف هؤلاء اللاعبين في هرمية اللاعبين Players. وفي الهرمية الفردية المـوجودة في شكـل 6.9 يمكننا أن نجمع كل هذه المعلومات بطريقة مباشرة أكثر.

وعلى هذا فأي الترتيبات يستخدم؟ ما نريـده في الواقـع هو أن نحصـل على كعكتنا وأن نأكلها كذلك. فنحن نود الاتصال الأكثر مـرونة وفي نفس الـوقت أقل

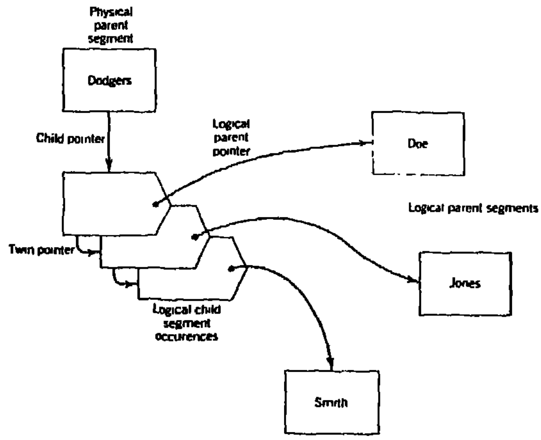
تكرار وأعلى درجة سلامة للبيانات بالنسبة للهرمية الفردية. في الواقع هناك تسهيلات في IMS تعطي أفضل ما يمكن من هذه الأمور وتسمى بالعلاقات المنطقية logical relationships.

يبين شكل 6.10 ما يعرف بعلاقة منطقية ذات اتجاه واحد. وقد استبدلنا جزء PLAYERS في هرمية الفرق Teams بنوع خاص من جزء مشير يعرف بأنه جزء منطقي تابع logical child segment. ويمكننا أن نقفز من الجزء المنطقي التابع بهذا في هرمية الفرق Teams إلى جزء PLAYERS في هرمية اللاعبين Players وإيجاد كل اللاعبين الموجودين في إحدى الفرق بكفاءة. كما يمكننا عمل ذلك مع جزء PLAYERS في هرمية الفرق Teams. في الواقع، هل حصلنا على كعكتنا ونستطيع أكلها كذلك؟. حيث تظل هرمية اللاعبين كما كانت فيمكننا أن نجد اللاعبين مباشرة (فهم زالوا جذراً لهرميتهم) ومعاملة حالة اللاعبين غير الموجودين حالياً في الفريق. وحيث أن معلومات اللاعبين تظهر في إحدى الهرميتين فقط فقد حذفنا تكرار اللاعبين الذي كان يقلقنا. أخيراً نستطيع باستخدام العلاقة المنطقية البدء عند جزء TEAM من هرمية الفرق Teams والعبور إلى هرمية اللاعبين Players مع انتقاء كل اللاعبين والاحصائيات السنوية ومعلومات المواقع التي نريدها. وهذا يأخذ في الاعتبار المستوى الأعرض لسلامة البيانات التي سبق أن تكلمنا عنها.

التنفيذ: كيف يعمل هذا في الواقع على قرص؟ يبين شكل 6.11 جزءاً من سجل قاعدة بيانات Dodgers من قاعدة بيانات Teams مع أجزاء الجذر



شكل 6.10
علاقة منطقية في اتجاه واحد



شكل 6.11
حدوث لعلاقة منطقية

لسجلات قاعدة البيانات Doe و Jones و Smith من قاعدة البيانات players. ويعمل الجزء المنطقي التابع مثل أي نوع من أنواع الأجزء في هرمية Teams. وعلى هذا فأول حدوث له تحت أي فريق محدد يمكن الوصول إليه عن طريق مشير تابع child pointer. ويتم الوصول الى البقية عن طريق سلسلة توأم كما هو مبين. ولكن يوجد الآن مشير إضافي موجود في الجزء المنطقي التابع (يسمى مشير عائل منطقي logical parent pointer).

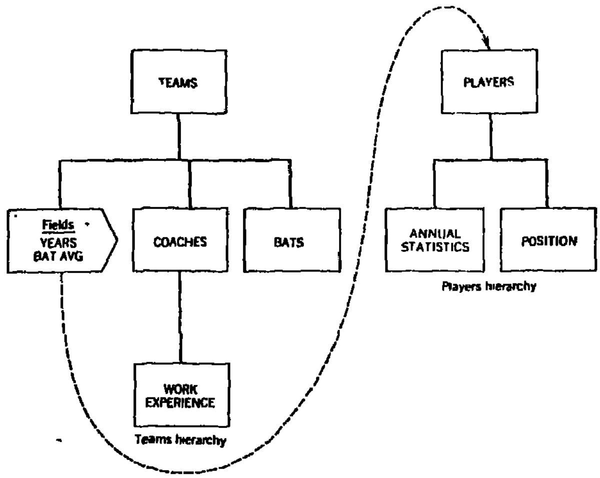
ويقفز الى قواعد البيانات الأخرى. ونظراً لأن الجزء المنطقي التابع هو في الواقع جزء من هرمية Teams فإننا نشير إلى جزئه العائل TEAMS على أنه عائله الطبيعي physical parent وإلى الجزء الذي يشير إليه في هرمية أخرى PLAYERS على أنه عائله المنطقي logical parent.

وكسؤال جانبي لماذا ندخل نوع جزء جديد، جزء منطقي تابع، للتأثير على علاقة منطقية؟ للذهاب من Dodgers الى الحدوثات المنطقية التابعة ومنها الى حدوثات PLAYERS هل نتتبع مشير واحد (المشير التابع من Dodgers) إلى مشير آخر (المشير المنطقي العائل) للحصول أخيراً على المقصد؟ لماذا لا يكون هناك مشير واحد في Dodgers يشير مباشرة الى لاعبيه؟ الإجابة هي أننا لا نستطيع عمل ذلك لأننا لا نعرف عدد لاعبي Dodgers' ولا نعرف المكان اللازم تركه لكل المشيرات لوضع جزء Dodgers. ووضع الأجزاء المنطقية التابعة كما فعلنا يسمح لأي عدد من اللاعبين ويجعل النظام يستخدم طرقه المعتادة في الاضافات والحذف ويبسط برمجة التطبيقات. كما يسمح ذلك بشيء يسمى بيانات التقاطعات «intersection data» كذلك.

بيانات التقـاطعات: سبق منـذ فترة طـويلة أن حددنـا موضـوع أن بيانـات الخواص البسيطة مثل مدير الفريق تخزن كحقل في جزء مناسب. ولكن هناك نوع آخر من بيانات الخواص عادة ما يكون هناك معلومـات وصفية لا يمكن أن تـوجد في أي من الأجزاء الخاصة بها. اعتـبر على سبيـل المثال عـدد السنوات التي لعبهـا أحد اللاعبين في فريق معين. هل تخـزن هذه المعلومـات في جزء الفـريق؟ لا لأنه هناك العديد من اللاعبين الموجودين في هذا الفـريق ولا يكون عملياً تخزين هـذه المعلومات في جزء الفريق. ماذا عن جزء اللاعب؟ إذا اعتبرنا الآن قاعدة البيانات هـذه لحفظ بيانـات تاريخيـة فيكون لـدينا نفس المشكلة فقـد يكون لعب الـلاعب للعديد من الفـرق المختلفة عـلى مدار السنـين والإجابـة هي التمتع بميـزة الأجزاء المنطقية التابعة. فأحد الأشياء التي يبينها شكل 6.11 هو أنـه هناك حـدوث مختلف للجـزء المنطقي التابع لكل فريق ولاعب مرتبطين. وعلى هذا فإن الحدوث المنطقي التابع الذي يصل حدوثيين لكينونتين هو المكان الطبيعي لتخزين مثل هذه البيانات المتقاطعة يبين شكل 6.12 كيف يمكننا أن نمثل البيانات المتقاطعة في تكوين عام. وبالمناسبة فإن هذه هي أول مرة نرى فيها حقولاً ـ فتمثل البيانات المتقـاطعة عـلى هيئـة حقول. وفي أي من هـذه السلسلة من الأشكال(BATAVG هـو متـوسط عمل الضربات للاعب محسوباً على مدار السنوات لعبه للفريق).

علاقات العديد للعديد: الهرميات كما سبق أن رأينا مناسبة جداً في معالجـة علاقة واحد للعديد. إلا أن الهرمية الفرديـة البسيطة لا يمكن أن تعـالج عـلاقات العـديد للعـديد. انـظر إلى شكل 6.5 مـرة أخـرى. كـل من المـدربـين Adams و Baker له خبرة تدريب من كلية College. دعنا نفترض في هذه اللحظة أنه لا تـوجـد حـقـول غـير حـقـل EXPERIENCE NAME في أجـزاء Work EXPERIENCE تجعل جزئي تدريب الكلية متكافئين. فإذا كان هذا هو الحال، أليس هناك تكرار عند حفظ جزء تدريب كلية تحت مدربين مختلفين؟

نعم هـذا صحيح. لكن مـا البـديـل؟ إذا كـان هنـاك كـل من Adams و Baker من خلال مشير تـابع أو مشـير توأم يشـير إلى جزء تـدريب كلية فـردي فسوف نحصل على خلط للكل سواء كان ذلك جزء تدريب Japan وجزء تـدريب Minor League المتصلة بجزء تدريب الكيـة عن طريق سلسلة تـوأم خاصـة بأي من Adams أو Baker. السبب الـذي يحدث هـو أن الموقف الـذي سبق وصفـه



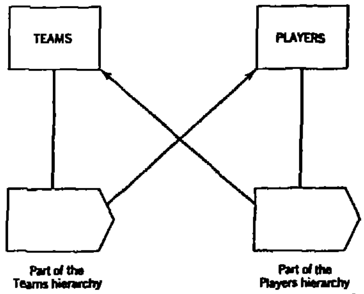
شكل 6.12
بيانات متقاطعة

يشمل في الواقع علاقة العديد للعديد. ونقول الآن أنه لأحد المدربين يوجد قطع عديدة من أعمال الخبرة ولكل قطعة من أعمال الخبرة يوجد العديد من المدربين الحاصلين عليها. وعلى هذا فلهذا السبب لا يمكن أن تعالج الهرمية البسيطة علاقة العديد للعديد.

وحل مشكلة علاقات العديد للعديد في IMS هي استخدام صورة أخرى من العلاقة المنطقية تسمى بعلاقة منطقية ذات اتجاهين bidirectional logical relationship. ويوضح شكل 6.13 اجزاء الهرميات Teams و Players كل منها معه تابع منطقي يشير إلى الآخر. اعتبر مرة أخرى أن البيانات تاريخية الطبيعة بحيث أننا مهتمون في حفظ تتابع كل اللاعبين الذين لعبوا لفريق معين. وهذه علاقة العديد للعديد والتي يمكن أن تنفذ في IMS بتكوين العلاقة المنطقية المزدوجة الاتجاه المبينة.

مرة أخرى بالنقل إلى منظر التخزين الواقع يبين شكل 6.14 حدوثاً للعلاقة المنطقية مزدوجة الاتجاه هذه. لقد كان في فريق Dodgers اللاعبين Doe و Jones كما أن Giants كان عندهم James و Smith. على العكس لقد لعب Doe لفريق Dodgers ولعب Jones لفريقي Dodgers و Giants وتعمل المشيرات التابعة والتوأم والعائلة المنطقية كما سبق ذكره ولكن في الاتجاهين.

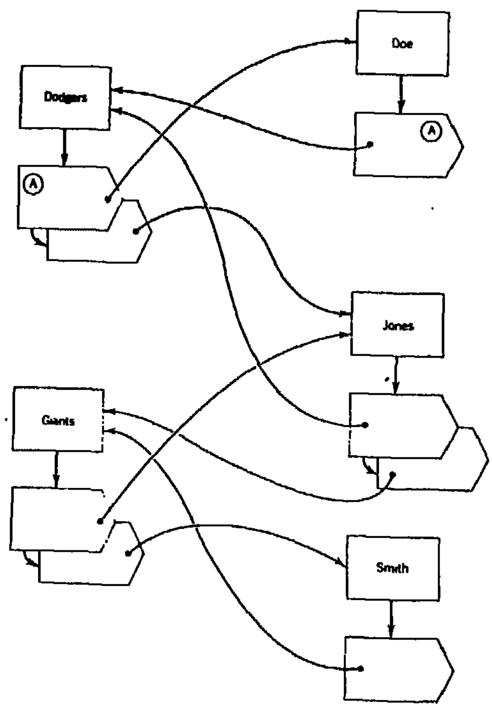
بهذا الترتيب يوجد حدوثان منطقيان تابعان لكل ملاءمة بين اللاعب والفريق خصيصاً واحداً منها تحت اللاعب المعنى والآخر تحت الفريق المعني. المستطيلان المسميان «A» في شكل 6.14 يصلان كل منهما Dodgers مع Doe.



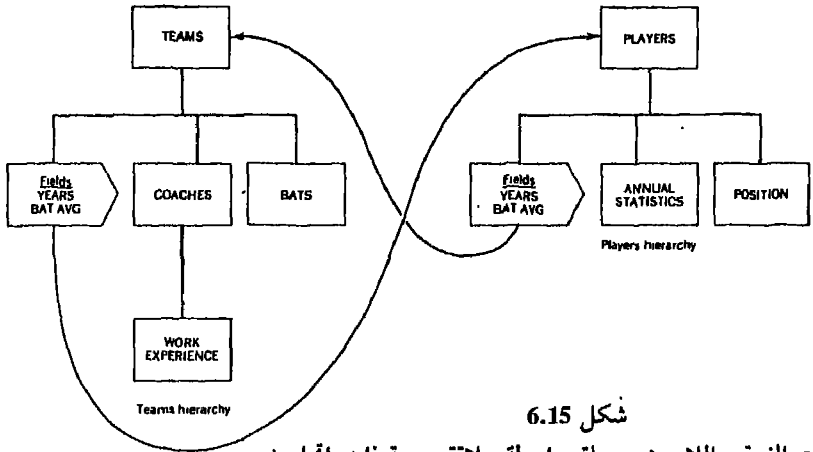
شكل 6.13
علاقة منطقية مزدوجة الاتجاه لعلاقة من العديد إلى العديد

سوف نلاحظ في المرور بذلك إذا وجدت بيانات متقاطعة مثل عدد السنوات التي لعبها اللاعب لفريق معين في هذا المثال فإنها تخزن في كل من الأجزاء المنطقية التابعة وإذا ما تغيرت في مكان واحد فيجددها النظام تلقائياً في المكان الآخر.

يبين شكل 6.15 محتويات هرميات Teams و Players متصلة بواسطة علاقة مزدوجة الاتجاه ازدواجاً طبيعياً physical paired ويشمل كذلك حقول YEARS و BAT AVG كعينة لبيانات متقاطعة. لاحظ أنه هناك صيغة أخرى أكثر تعقيداً بأنها مزدوجة افتراضياً في تكوينها للعلاقة المنطقية مزدوجة الاتجاه تسمى «virtually paired» توجد فيها البيانات المتقاطعة في أحد جوانب العلاقة فقط.



شكل 6.14
حدوث لعلاقة منطقية مزدوجة الاتجاه



شكل 6.15
هرميات الفرق واللاعبين متصلة بواسطة علاقة هرمية ذات اتجاهين

قواعد البيانات المنطقية والطبيعية:

ألم يكن غريباً منذ فترة مضت أن يسمع حديث عن العائل الطبيعي والعائل المنطقي لجزء منطقي تابع؟ هل كنا نقول أن الجزء المنطقي التابع يمكن أن يكون له نوعي اجزاء عائلة أحدهما طبيعي والآخر منطقي؟ ألا يكون هذا عكس الحقيقة بأنه في الشجرة يمكن أن يكون للعقدة عائل واحد فقط؟ نعم ولا.

التكوينان المنفصلان في شكل 6.8 هما هرميتان واضحتان. والتكوينات المركبة في أشكال 6.10 و 6.13 و 6.15 هي شبكات مقيدة تقنياً. سوف نناقش الشبكات كشكل لتخزين قواعد البيانات في الفصل السابع . ومن الكافي القول الآن أن الشبكة صورة أكثر تعقيداً من الهرمية (في الواقع الهرمية هي نوع محدود من الشبكة). المهم بالنسبة لذلك في اصطلاحات IMS هو أنه أثناء استخدام العلاقات المنطقة ينتج من الناحية التقنية شبكات في التكوين الطبيعي الأساسي ودائماً ما يكتب المبرمجون براجهم بعناية على أساس الهرميات.

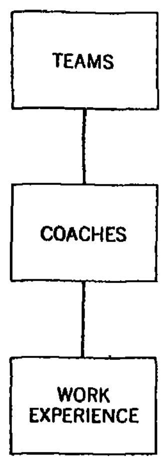
كل تكوينات قواعد بيانات IMS التي شاهدناها حتى الآن تعرف بأنها قواعد بيانات طبيعية physical database لنظام IMS. ويكتب المبرمجون (سوف نلقي نظرة سريعة على برمجة IMS) شفرتهم على أساس قواعد بيانات منطقية physical database لنظام IMS. وقواعد البيانات المنطقية هي في الواقع أوصافاً لما يجب أن يعتبره المبرمجون لأشكال البيانات. أجزاء قواعد البيانات المنطقية ترسم على أجزاء قواعد البيانات الطبيعية التي تحتوي على البيانات الفعلية. قاعدة

بيانات منطقية لنظام IMS في أبسط الحالات هي شجرة جزئية لهرمية طبيعية لنظام IMS أي أنها تحتوي على عقدة وبعض أو كل أفرعها. عقدة البداية أي جذر قاعدة البيانات المنطقية يجب أن تكون جذراً لقاعدة البيانات الطبيعية أو جذراً له فهرساً ثانوياً (سوف يناقش فيها بعد) يشير إليها. شكل 6.16 هو قاعدة بيانات منطقية مستخلصة من قاعدة البيانات الطبيعية Teams. قارن شكل 6.16 مع شكل 6.8 لترى الرسم والعلاقة بين الشكلين. وشكل 6.17 هو قاعدة بيانات منطقية مستخلصة من الهرميات الطبيعية Teams و Players والعلاقة المنطقية بينهما. قارن بين شكل 6.17 وشكل 6.10 لترى كيفية توصيل العلاقة المنطقية لقاعدتي بيانات طبيعيتين وكيف تصبح فرعاً في قاعدة بيانات منطقية.

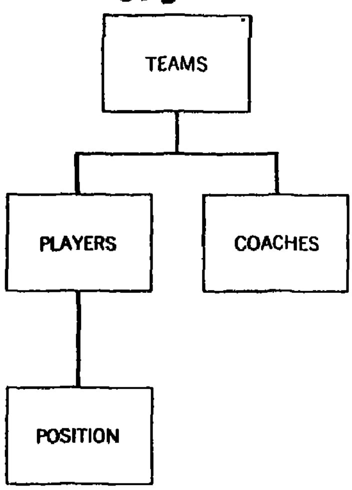
هناك ميزتان لمفهوم قواعد البيانات المنطقية. الأولى هي البساطة في أن المبرمجين يعملون دائماً مع هرميات وليس مع شبكات. والثانية هي المقدرة على وجود بيانات اتصال ببرامج مختلفة من نفس قاعدة البيانات الطبيعية مع تقييد كل برنامج للأجزاء التي يحتاجها (وحتى الحقول التي يحتاجها) فقط.

طرق الاتصال:

لقد ناقشنا في الفصل الثالث طرق اتصال بملفات خطية بسيطة. وقد ظهر حتى الآن في هذا الفصل التعبير التالي (بافتراض أنه يمكننا أن نصل إلى حدوثات



شكل 6.16
قاعدة بيانات منطقية
مستخلصة من هرمية طبيعية فردية



شكل 6.17
قاعدة بيانات منطقية مستخلصة
من هرميتين متصلتين بعلاقة منطقية

الجذر مباشرة) العديد من المرات. وقد آن الأوان لمعرفة معنى لهذا التعبير ومناقشة طرق الاتصال في محتوى الهرمية.

هناك أربع طرق اتصال رئيسية في نظام IMS وهي:

١ ـ طريقة اتصال هرمية تتابعية. Hierarchic Sequential Access Method (HSAM).

٢ ـ طريقة اتصال هرمية تتابعية مفهرسة. Hierarchic Indexed Sequential Access Method (HISAM).

٣ ـ طريقة اتصال هرمية مفهرسة مباشرة. Hierarchic Indexed Direct Access Method (HIDAM).

٤ ـ طريقة اتصال هرمية مباشرة. Hierarchic Direct Access Method (HDAM).

ويحدد اختيار طريقة اتصال IMS شيئين في الواقع أحدهما هو ما إذا كانت سجلات قاعدة البيانات الفردية ستخزن باستخدام تتابع أو مشير مبنى على تقنيات التخزين. والثاني هو نوع الاتصال الذي يعده النظام لحدوث الجذر لكل سجل من سجلات قاعدة البيانات. ويسرد شكل 6.18 كل الامكانيات.

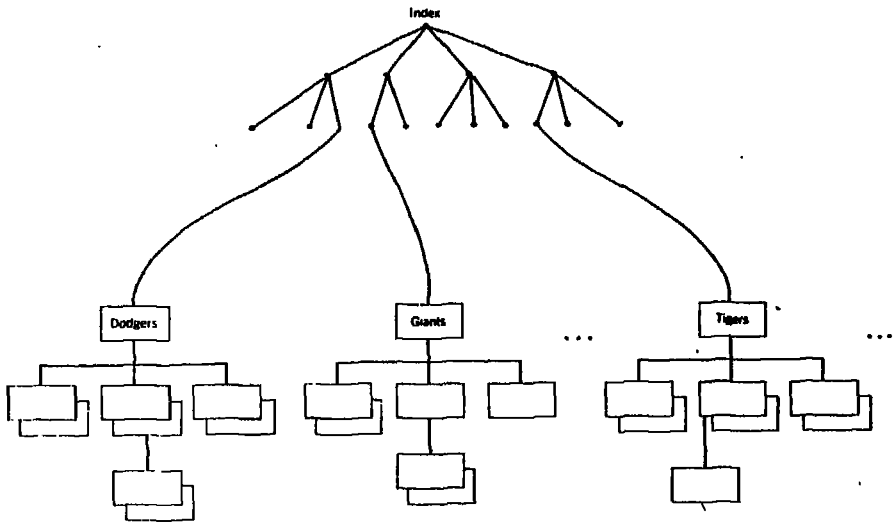
	Access to Root	Database Record Structure
HSAM	Sequential	Sequential
HISAM	Indexed	Sequential
HIDAM	Indexed	Pointer-Based
HDAM	Hashed	Pointer-Based

شكل 6.18
طرق اتصال IMS

في اتصال HSAM تعد سجلات قاعدة البيانات باستخدام تقنية تخزين تتابعية. ومع حدوث اتصال بالجذر لا توجد مقدرة اتصال مباشر. وتخزن سجلات قاعدة البيانات ببساطة واحداً بعد الآخر في تتابع على قرص طبقاً لحقل رئيسي في جزء الجذر. ومع غياب أي إمكانية للاتصال المباشر يستخدم اتصال HSAM. مثال ذلك عند تخزين الأرشيف.

وكما هو الحال في HSAM فتعد سجلات قاعدة البيانات في HISAM

بـاستخدام تقنيـة التخزين التتـابعي. ولكن في HISAM يمكن الاتصـال مبـاشرة بسجلات قاعدة البيانات عن طريق فهرس يشير إلى حقل رئيسي لحدوثـات الجذر شكل 6.19. والفهرس المستخدم حاليـاً عادة مـا يكون فهـرس VSAM - KSDS الذي سبق مناقشته في الفصل الثالث. وحيث أن هذا التكـوين هو في واقـع الأمر تكوين تتابعي مفهرس فلا يكون الاتصال بالسجلات مباشراً فقط بـواسطة مفتـاح جزء الجذر ولكن تخزين السجلات كذلك في تتابع طبيعي طبقاً لهذا الحقل أيضاً.



شكل 6.19
مفهوم الفهرس من HISAM أو HIDAM

أما في HIDAM فتعد قاعدة البيانات باستخدام مشـير تابـع وترتيب مشـير توأم (أو أي طريقة معروفة على أنها مشـيرات هرميـة). ويستخدم هنـا مثل حـالة HISAM فهرس VSAM - KSDS للاتصـال بالجـذور. وخليط الاتصال المبـاشر والاتصال المفهرس بالجذور بالإضافة الى التكوين المبني على المشير يجعل HIDAM مرشحاً للاستخدام في نظام الخط المفتوح والذي يتطلب استجابة سريعة.

وتعد سجلات قـاعدة البييـانات في HDAM بـاستخدام طـريقة مبنيـة على مشير. ويمكن الاتصال مباشرة بالجذور ولكن باستخدام طريقـة الاتصال المعتمـدة على التكرار في هذه الحالة. وهذا يعني إن الاتصـال في نظام HDAM بحـدوثات الجذور يكون أسرع عادة منه في نـظام HIDAM. ولكن نظراً لاستخـدام التكرار

hashing فتكون سجلات قاعدة البيانات في ترتيب معين سواء كان تتابعياً أو غيره على القرص. وتعتبر طريقة الاتصال الأسرع هي HDAM بصفة عامة.

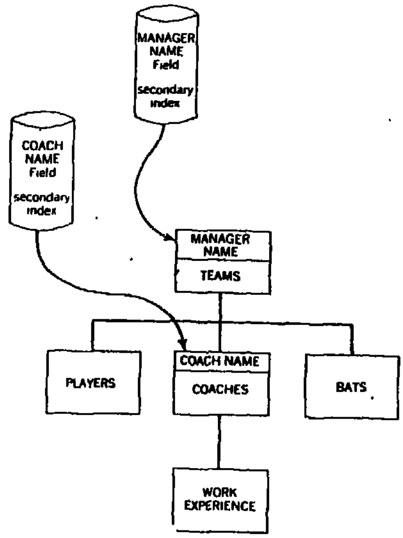
هناك ملاحظة أخيرة على هذا الموضوع فشكل 6.19 هو شكل أكثر عمقاً فهو يبين استخدام مفهوم الشجرة لكل من تكوين الشجرة وتخزين البيانات في نفس الوقت. فهرس VSAM - KSDS مبني على مفهوم الشجرة وكذلك طريقة تخزين البيانات الفعلية في سجلات قاعدة بيانات IMS. والتمييز بين هذين الاستخدامين للشجرة وحقيقة أنه يمكن استخدامها ترادفياً مهمة.

فهارس ثانوية:

باستخدام طرق اتصال IMS يمكن تحقيق اتصال بسجلات قاعدة البيانات من خلال الحقل الرئيسي لجزء الجذر. إلا أنه توجد ظروف عديدة يكون مطلوباً فيها اتصال مباشر إضافي. ويمكن أن يأخذ هذا صورة اتصال مباشر لجزء الجذر عن طريق حقل بدلاً من مفتاح أو اتصال مباشر بأي حقل في أي جزء آخر. مثال ذلك بالعودة إلى هرمية Teams في شكل 6.8 قد نريد اتصال مباشر بجزء Teams عن طريق حقل MANAGER NAME أو اتصال مباشر بجزء COACHES عن طريق حقل COACH NAME.

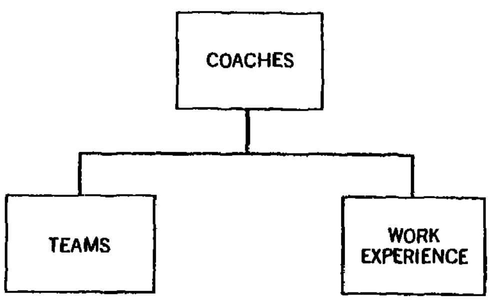
مثل هذا الاتصال المباشر الإضافي يمكن أن يتحقق باستخدام مفتاح ثانوي secondary index. وبينما يجب أن تكون قيم الحقول الرئيسية المستخدمة في طرق الاتصال فردية فتحتاج الحقول التي تعد على أساسها الفهارس الثانوية أن تكون فردية أيضاً. يبين شكل 6.20 هرمية Teams بفهارس ثانوية تشير الى حقلين مقترحين. الفهارس الثانوية نفسها منظمة كفهارس VSAM - KSDS إضافية.

هناك تأثير جانبي لاستخدام الفهارس الثانوية وهو المرونة الإضافية في إعداد قواعد البيانات المنطقية. لقد سبق أن أشرنا أنه بدون استخدام الفهارس الثانوية يجب أن يكون جذر قاعدة البيانات المنطقية هو جذر قاعدة البيانات الطبيعية. وعلى أية حال فباستخدام الفهارس الثانوية يمكن أن يكون جذر قاعدة البيانات المنطقية هو جذر قاعدة البيانات الطبيعية أو جزء يشار إليه بفهرس ثانوي. ويمكن القول بأن جذر قاعدة البيانات المنطقية يجب أن يرسم على جزء



شكل 6.20
الفهارس الثانوية

طبيعي يمكن الاتصال به مباشرة. إذا كان جـذر قاعـدة البيانـات عبارة عن جـزء يشار إليه بفهرس ثانوي .فيجب على تكوين قاعـدة البيانـات المنطقي أن يتفق مـع قواعد معينة. يبين شكل 6.21 قاعدة بيانـات منطقيـة مبنية عـلى قاعـدة البيانـات الطبيعية الموجودة في شكـل 6.20. واعتماداً عـلى الفهرس الثـانوي Coach Name يعد جزء COACHES كجذر للهرمية المنطقيـة وتقع الأجـزاء التابعـة في الهرميـة المنطقية في فئتين. فيمكن أن تكون سلالات في الهرميـة الطبيعيـة اجزاء WORK EXPERIENCE. أو يمكن أن تكـون أجزاء في مسـار فـردي لأجـزاء من الجـزء المفهرس COACHES والذي يعـود للخلف الى قاعـدة البيانـات الطبيعيـة (والتي تؤهل جزء TEAMS فقط في هذا المثال). في هذا الترتيب يستخدم النظام مشيراً اختيارياً اسمه مشير العائل الـطبيعي physical parent للإشـارة لأعلى في الهـرمية الـطبيعية من COACHES إلى TEAMS عـلى سبيل المثـال في شكل 6.20 بحيث أن COACHES يمكن أن يكون له نوع جزء TEAMS في شكل 6.21.



شكل 6.21
قاعدة بيانات منطقية مبنية على فهرس ثانوي

دعنا نلاحظ في تمرير ذلك أنه حيث أن الفهرس الثانوي يرتب قيم الحقل الذي يفهرسه فيمكن استخدام الفهرس الثانوي في الاسترجاع التتابعي كذلك. مثال ذلك يمكن استخدام حقل COACH NAME وهو فهرس ثانوي في شكل 6.20 في استرجاع كل المدربين في ترتيب أبجدي بغض النظر عن الفريق الذي يعمل فيه المدرب. والمشكلة في هذا هي الأداء حيث أن المدربين مخزنين بالنسبة للفرق التي يعملوا لها فإن مخازنهم قد أعدت بغض النظر عن حفظ أسماء المدربين المتقاربة في الترتيب الأبجدي بجوار بعضها البعض على القرص. واستخدام الفهرس الثانوي لاسترجاعهم تتابعياً يتسبب في حركة كبيرة لذراع حركة اتصال القرص وينتج عن ذلك أداء ضعيف.

مجموعات التحكم:

نحن على وشك مناقشة كيفية استخدام المبرمجين لنظام IMS وبياناته (لغة معالجة البيانات data manipulation language) إلا أنه يجب أن نذكر في البداية الاتصال داخل نظام IMS بين تكوينات البيانات وبرامج التطبيقات التي سوف يستخدمها.

ومن الجيد والمفيد الحديث عن الهرميات والعلاقات المنطقية وما إلى ذلك وعن كيفية تخزينها والاتصال بها. لكن كيف يعرف نظام IMS كقائم بدور الرجل الوسيط بين البيانات والبرامج طبيعة البيانات وكيفية تخزينها. والإجابة هي مع مجموعة من التعليمات المعطاة لنظام IMS تعرف بأنها مجموعات التحكم control blocks.

هناك أنواع عديدة من مجموعة التحكم في نظام IMS ومن المهم رؤية كيفية اتصالها بالمفاهيم والتكوينات التي سبق أن تكلمنا عنها. إحداها يسمى وصفاً طبيعياً لقاعدة البيانات Physical Data Base Description (Physical DBD) ويصف DBD الطبيعي تكوينات التخزين الطبيعي بما في ذلك الأجزاء وكيفية ترتيبها في هرمية والحقول الرئيسية والعلاقات المنطقية والفهارس الثانوية وطرق الاتصال بنظام IMS والاختيارات المتاحة للمشيرات. ومن المهم معرفة الطريقة التي يتحدد بها تكوين الهرمية بتحديد العائل لكل جزء موجود في القائمة. وحيث أن إحدى خواص التكوين الشجري هو أن كل عقدة لها عائل واحد فقط، فيما

عدى الجزء الـذي ليس له أي عـائل، فـإن هذه الـطريقة هي طـريقة غـير مبهمة لتحديد التكوينات الشجرية للنظام.

وعـلى هذا فهنـاك عدة أنـواع من مجموعـات التحكم التي تصف ما نسميـه قواعد البيانات المنطقية أي الطريقة التي يمثـل المبرمجـون بها البيـانات. أولاً هنـاك مجموعة تحكم الـوصف المنطقي لقـاعدة البيـانات Logical Data Base Description (Logical DBD) ويختـار أجزاء من عـلاقة فـردية أو منـطقية متصلة بهـرميـة طبيعية تمثل وجهـة نظر المـبرمج الهـرمية. مجمـوعة اتصـالات البرنـامج Program Communication Block (PCB) هي مجموعة تحكم تستخلص اشجاراً جزئية من LBS المنطقية (القـواعد التي يمكن أن تجعـل الأجزاء جـذوراً في تكوينـات هرميـة تسري هنا كذلك) لمستفيدين معينين من برنامج معين. ويرتبط أي بـرنامـج IMS بمجموعة Program Specification Block (PSB) واحدة تسرد كل PCB اللازمة لهذا البرنامج.

السطح البيني للغة DL/I: DL/I Language Interface

مقدمة:

بالنسبة لاصطلاحات معالجة البيانات فإن DL/I هو مـا سبق أن أشرنا إليـه بـاللغة المحتـواه embedded. عبارات DL/I التي تصـل الى البيانـات في قـاعـدة بيـانات IMS هي محتـواه في برامـج تـطبيقـات الكـوبـل و PL/I ولغـة المجمـع. وبالنسبة للتأثير تعمل عبارات DL/I مثل الماكرو أو البرنامج الفرعي المرتبط بنـظام ادارة قاعدة بيانات IMS معطياً تعليـمات لاسترجـاع أو لتخزين بيـانات في قـواعد بيانات IMS. دعنا ننظر إلى كيفية عمل ذلك.

آلية الاستدعاء:

تحتوي عبارة استدعاء DL/I عـلى عدد من المؤشرات كـل منها مصمم لأداء وظيفة محددة في الاتصال بين البرنامج وقاعدة البيانات أثناء عمليـة نقل البيـانات. والمؤشر الذي يحدد نبرة عبارة الاستدعاء هذه هو «Function Code» والذي يحدد نـوع عبارة نقـل البيانـات. ويميز أربعـة أنواع أسـاسية لمثـل هـذه العمليـات (١) اسـترجـاع و(٢) تجـديـد و(٣) ادخـال و(٤) حـذف خـلال بعض الخليط لهـذه

العمليات كما سنرى ذلك.

وبمجرد تحديدك نوع العملية المطلوب أداؤها يجب عليك أن تحدد جزء قاعدة البيانات التي تعمل عليه. ولهذا الغرض تشمل عبارة استدعاء DL/I اسم PCB. تذكر أن PCB هي مجموعة تحكم تعرف جزء من قاعدة البيانات من وجهة نظر المبرمج.

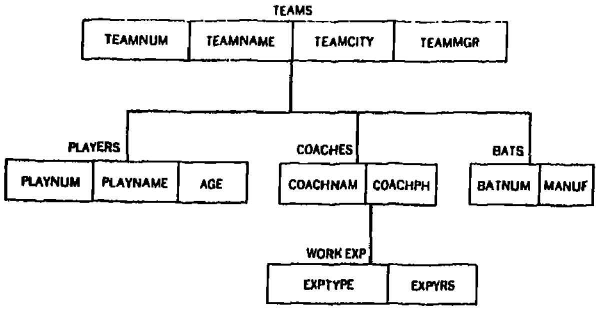
كما يجب أن يوجد كذلك منطقة تخزين في الذاكرة الرئيسية تعمل كنقطة نقل فعلية للبيانات. وعندما يتم تحديد عملية استرجاع أي معرفة البرنامج المكان الذي يضع فيه البيانات بعد أن يجدها في قاعدة البيانات.

كما أنه في حالة عملية الإدخال أو التجديد يكون هذا هو المكان الذي يضع البرنامج البيانات الجديدة فيه للنظام لكي يستخلصها النظام وينقلها لقاعدة البيانات. ويسمى هذا في DL/I بمنطقة مدخلات ومخرجات I/O Area.

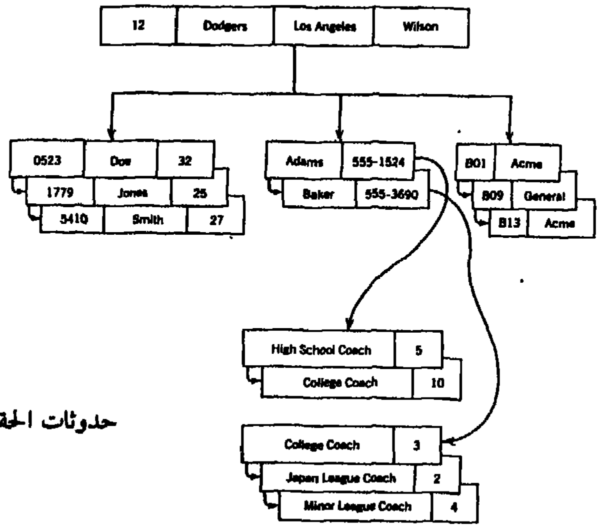
بالإضافة إلى ذلك هناك معلومات أخرى يمررها النظام للخلف الى المستفيد من البرنامج بعد عمل استدعاء. ويشمل ذلك قائمة بحدوثات الحقل الرئيسي للأجزاء المطلوبة. وهذه القائمة توضع في منطقة تخزين محددة (تسمى منطقة تغذية مرتجعة رئيسية Key Feedback Area). كما يرسل النظام كذلك رمز حالة للخلف مقدماً معلومات إضافية عن نتيجة الاستدعاء.

وأخيراً وليس آخراً في عبارة الاستدعاء توجد قوائم بحث للجزء Segment Search Argument (SSA). وهي عبارة عن مؤشرات توجه النظام خلال الهرمية المحددة بواسطة PCB وذلك إلى الموقع الذي تحدث فيه العملية المطلوبة. وسوف نرى توضيحاً أكثر في المثال التالي لكل من هذه المؤشرات والمفاهيم.

شكل 6.22 هو وصف تفصيلي أكثر لهرمية Teams الموجودة في شكل 6.4. وفي هذه الحالة فهي قاعدة بيانات منطقية تشمل محتوى قاعدة البيانات الطبيعية Teams. ولا يبين الشكل الأجزاء الموجودة في الهرمية فقط ولكنه يبين كذلك الحقول الموجودة في الأجزاء. وكل من أسماء الأجزاء والحقول مختصرة بطريقة تناسب الاستخدام في عبارات البرنامج. والحقل الموجود على أقصى يسار كل جزء هو حقل رئيسي فريد. وفي DL/I يجب أن يكون لجزء الجذر حقل رئيسي فريد إلا أن الأجزاء التابعة قد يكون أو لا يكون لها مثل هذا الحقل الرئيسي. وشكل 6.23



شكل 6.22
هرمية الفرق
بين أسماء أجزاء وأسماء حقول



شكل 6.23
حدوثات الحقل في سجل قاعدة بيانات Dodgers

هو حدوث هرمية Teams لـ Dodgers (سجل قاعدة البيانـات Dodgers) المبين في شكل 6.5 لكنه يشمل في هذه المرة حدوثات الحقل. وكما هو مبـين فهو مكـون من مشـيرات معتمـدة وتـواءم. ونفـترض أن طـريقـة IMS لـلاتصـال هي إمـا HIDAM أو HDAM بحيث أن الحقل الرئيسي لجزء الجذر أما أن يكون مفهرساً أو متكرراً بالنسبة للإتصال المباشر.

الاسترجاع المباشر:

أبسط نوع لعبارة DL/I هو عبارة الإحضار الفردي Get Unique والتي لهـا رمـز وظيفي GU وهي مصممة لإيجـاد حدوث جـزء محدد مبـاشرة والعـودة بـه أو بالعديد من حقولـه إلى البرنـامج المنـادي. وطبيعة الاسـترجاع المبـاشرة مبنية عـلى توقيع جزء الجذر للهرمية المنطقية. مرة أخرى، يعني ذلـك أن هذا الجـزء أما أنـه يرسم لجزء عبارة عن جذر للهرمية الطبيعية (ويمكن الاتصال به بطريقة تكرارية أو مفهرس) أو أنه جزء له فهرس ثانوي يشير إليه. دعنا ننـظر إليه بـالنسبة إلى مثـال البيسبول.

إفرض أنك تريد إيجاد حدوث جذر لسجل قاعدة البيانات Dodgers. لأنك تريد إيجاد اسم المدرب والمدينة التي يقطن بها افرض أنك تعرف أنه عند العقدة التي تريد منها المعلومات أن Dodgers هو الفريق رقم 12. في الموقع المناسب من منطق البرنامج فإنك تصدر استدعاء DL/I التالي: (لاحظ أن للبساطة أن كل من الأمثلة التالية يبين رمز وظيفي وقوائم بحث جزء فقط).

GU TEAMS (TEAMNUM = 12)

يسمى التعبير «TEAMS (TEAMNUM = 12)» بـ SSA المؤهل. تذكر أن PCB المحددة في الاستدعاء (غير الموضحة في هذا المثال) تحدد أي هرمية تريدها أنت. اسم الجزء TEAMS في SSA يحدد نوع الجزء في هذه الهرمية التي تهتم بها. اسم الحقل وقيمة TEAMNUM = 12 في SSA تحدد أي حدوث لنوع الجزء هذا في الهرمية التي تهمك. وعلى هذا فالتعبير «GU TEAMS (TEAMNUM = 12)» مع هرمية Teams المحددة في PCB يقول للنظام «اذهب إلى هرمية Teams وأوجد حدوث جزء TEAMS الذي له قيمة حقل TEAMNUM مساوية 12 ثم عد بهذا الجزء الى البرنامج». في هذه الحالة حيث أن الحقل الموجود في SSA (حقل البحث) هو حقل رئيسي فردي فعلى هذا بالطبع يكون هناك جزء واحد فقط مشمول.

وحيث أن TEAMNUM هو حقل رئيسي فريد وفي واقع الأمر فإنه هو الحقل الرئيسي الجذر فيكون هذا النداء مباشر جداً. ويحدد النظام طريقة اتصال IMS التي خزنت بها قاعدة البيانات ويحدد كذلك ما إذا كانت سجلات قاعدة البيانات مخزنة بفهرس معتمداً على مفتاح الجذر (TEAMNUM) أم أنها مخزنة مباشرة بطريقة تكرارية مبنية على مفتاح الجذر (TEAMNUM) ويجد النظام جزء TEAMS لـ Dodgers ويعيد:

12 Dodgers Los Angeles Wilson

لمنطقة المدخلات والمخرجات المحددة.

كما يمكنك أن تصدر كذلك نداء Get Unique بحقل بحث ليس مفتاحاً لجزء الجذر (وليس حقلاً حدث له فهرسة بفهرس ثانوي). وفي هذه الحالة يجري

النظام بحثاً تتابعياً خلال سجلات قاعدة البيانات في قاعدة البيانات هذه بحثاً عن الجزء المطلوب. ومن الواضح أن هذه العملية تستغرق وقتاً كبيراً وعلى هذا فيجب أن يأخذ تصميم قاعدة البيانات وكذلك استدعاءات DL/I هذا في الحسبان.

وفيما يلي مثال به العديد من SSA. افرض أنك تعرف أن Baker وهو مدرب في Dodgers اعتاد أن يكون مدرباً للناشئين وإنك تريد أن تعرف عدد السنوات التي قضاها في تدريب الناشئين. يمكن أن تصدر النداء DL/I التالي:

```
GU TEAMS (TEAMNUM = 12)
   COACHES (COACHNAM = Baker)
   WORKEXP (EXPTYPE = Minor League Coach).
```

انظر إلى شكل 6.22 ولاحظ أن كل من الثلاثة SSA يشير إلى نوع سجل في مستوى متتالي لأسفل في الهرمية. مرة أخرى حيث أن أول SSA يشير إلى الحقل الرئيسي لجزء الجذر فيجد النظام جزء الجذر مباشرة. وبعد ذلك باتباع التعليمات الموجودة في SSA التالي فإنه ينتقل من سجل قاعدة بيانات Dodgers متتبعاً المشيرات التابعة والتوائم لحدوث Baker في COACHES ثم حدوث Minor League Coach في WORKEXP. لاحظ أنه إذا لم يكن أي من حقول البحث في مستوى SSA الأقل فردياً فإن النظام يختار أول حدوث الذي يحقق الشرط داخل قيود أعلى مستوى SSA. ويعود بآخر جزء مسمى في قائمة SSA الى منطقة المدخلات والمخرجات.

Minor League Coach 4

وتحتوي منطقة التغذية المرتجعة الرئيسية على ما يلي:

12 Baker Minor League Coach

وهي المفاتيح الرئيسية للثلاثة أجزاء المحددة بواسطة SSA ويتم الاتصال بها أثناء البحث. قيمة حقل EXPYRS في الجزء المسترجع هي 4 وهذه هي إجابة السؤال.

كما يمكن أن تكون SSA غير مؤهلة كذلك. افرض أننا نعرف أن كل مجموعة من مجموعات خبرة المدربين مرتبة من أقدم خبرة لأحدثها وإنك تريد

معرفة أقدم خبرة للمدرب Baker دون أن تعرف مسبقاً نوعها. يمكنك أن تصدر الأمر التالي:

```
GU TEAMS (TEAMNUM = 12)
    COACHES (COACHNAM = Baker)
    WORKEXP
```

يبحث النظام عن Baker تحت Dodgers ويستخلص أول جزء من خبرته دون تأهيل.

ماذا يحدث إذا أصدرت Get Unique ولم يوجد الجزء الذي حددته في SSA؟ هنا يأتي دور رمز الحالة. فلكل رمز عمل DL/I يوجد عدد من رموز الحالة يمكن أن يعاد بواسطة النظام بعد إصدار نداء بهذا الرمز. في حالة الجزء المفقود المذكور هنا يعود رمز حالة محدداً «Not Found» غير موجود. وبعد كل نداء DL/I يجب أن يشمل البرنامج تعليمات لاختبار رموز الحالة التي تمثل امكانيات مناسبة لرمز الوظيفة المستخدم في الاستدعاء.

الاسترجاع التتابعي:

نظام IMS قادر كذلك على عمل أبحاث تتابعية في قاعدة البيانات. وفي الصورة المبسطة لهذه البحوث التتابعية فإنها تحتوي على استرجاع كل أو بعض الأجزاء الموجودة على سلسلة توام. وعلى أية حال يمكن أن تشمل البحوث المتتالية حدوثات لأنواع اجزاء مختلفة في سجل قاعدة البيانات وكذلك حدوثات أجزاء في أكثر من سجل قاعدة بيانات واحد. ومن المهم ملاحظة أنه عندما يحدث الأمر الأخير يكون ترتيب الاسترجاع تتابعياً سابق الترتيب: من القمة إلى القاعدة ومن اليسار الى اليمين ومن الأمام الى الخلف بغض النظر عما إذا كان التخزين الفعلي لحدوثات الجزء يتحقق واقعياً بأنه تتابعياً في تتابع سابق الترتيب أم أنه يتحقق بمشيرات تابعة وتوائم.

وفي مواقف الاسترجاع المباشر التي رأيناها لتونا دائماً ما يبدأ البحث عند بداية قاعدة البيانات. في الاسترجاع التتابعي سوف نعد موضعاً عند سجل معين في قاعدة البيانات ونعمل من عنده. هذا الموضع الذي يمكن تحديده بنداء GU أو

بأحد أنواع النداءات التتابعية التي سوف نفحصها الآن، يحفظ بمشيرات «الوضع الحالي» الذي يكون النظام مسؤولاً عن حفظه. وهذا يعني أن النظام يحفظ دائماً تتبع موقعه الدقيق من قاعدة البيانات في أي وقت كان.

وكملاحظة أخيرة قبل فحص عدة أمثلة دعنا نعتبر كيفية استخدام البرنامج لبيانات تعود إليه كنتيجة لاسترجاعات تتابعية من IMS. في حالة الاسترجاع المباشر يصدر البرنامج أمراً فردياً بطلب حدوث جزء فردي والعودة به. وفي صورة الخط الداخلي in - line يمكن لمنطق البرنامج أن يفعل أي استخدام للبيانات مفترض أن يؤديه. أما في حالة الاسترجاع التتابعي فيجب أن يخطط البرنامج لتشغيل حدوثات متعددة لنفس نوع الجزء (على سلسلة توأم) أو حتى حدوثات لأنواع اجزاء مختلفة. وعلى هذا فنداءات IMS التي تشمل عبارات استرجاع تتابعي يجب أن تكون في منتصف أحد أنواع الدورات في منطق البرنامج لتشغيل حدوث واحد ثم حدوث آخر وحدوث آخر وهكذا حتى يتم تشغيل كل الحدوثات المطلوبة.

إفرض أنك تريد استرجاع كل الحدوثات لكل أنواع الأجزاء تحت جذر Dodgers يمكنك أن تصدر أولاً الأمر:

GU TEAMS (TEAMNUM = 12)

والذي يمكن أن يحقق شيئين. أول شيء يمكنه أن يحصل على المعلومات الموجودة في حدوث أجزاء الجذر هذا. وثانياً يمكنه أن يعد مشيراً الموقع الحالي للنظام عند حدوث هذا الجزء. بعد ذلك للحصول على كل حدوثات الأجزاء التابعة فعليك أن تصدر عبارة Get Next في إحدى أنواع دورات البرنامج.

GN

وفي كل مرة تنفذ عبارة الاستدعاء البسيطة جداً GN هذه في الدورة يعيد النظام جزءاً آخراً في ترتيب تتابعي سابق الترتيب. أولا يكون هناك جزء PLAYERS الخاص بـ Doe ثم الخاص بـ Jones ثم الخاص بـ Smith ثم جزء COACHES الخاص بـ Adams يليه جزئي WORKEXP الخاصين به ثم Baker يليه الثلاثة أجزاء WORKEXP الخاصة به ثم الثلاثة أجزاء BATS. إذا

نفذ نداء GN مرة ثانية فإن حدوث الجزء التالي الذي يعاد يكون جزء الجذر لسجل قاعدة البيانات التالي. ولإيقاف حدوث ذلك يجب أن يختبر البرنامج بصفة مستمرة رمز الحالة الذي يعيده النظام. وبعد نداء GN يحدد النظام إذا كان مشير الوضع الحالي قد انتقل مستوى لأعلى في سجل قاعدة البيانات أو أنه استرجع نوع جزء مختلف عن آخر واحد أو أنه وصل إلى نهاية قاعدة البيانات. كما يمكن للبرنامج أن يختبر بالطبع نوع الجزء المسترجع الذي يجيب كذلك على السؤال عما إذا كان قد ذهب إلى سجل قاعدة البيانات التالي (أي إذا ما كان نوع الجزء المسترجع هو الجذر) أم لا.

مثال ذلك وكاختلاف عن هذا الموضوع إذا ما أصدرت ما يلي:

GN COACHES

في دورة فإنك تسترجع كل حدوثات أجزاء COACH بدءاً من النقطة الموجود عندها مشير الوضع الحالي وحتى النقطة التي ينهي عندها برنامجك تنفيذ الدورة.

وبينما تكون عبارة GN مفيدة بوضوح فقد تؤثر عليك بأنها تبدو غير محكمة في الطريقة بأنها يمكنها أن تقفز من نوع جزء لنوع جزء آخر. وما هو مطلوب هو طريقة لتحديد مدى نداء الاستدعاء التتابعي بعناية أكثر. وفي الواقع من الشائع جداً وجود مواقف تطبيقات نريد التركيز فيها على حدوث جزء خاص ونعتبر الحدوثات التابعة له فقط دون أي حدوثات سجلات أخرى في قاعدة البيانات. لتحقيق هذا في DL/I يوجد رمز وظيفي آخر يسمى Get Next Within Parent احصل على التالي تحت نفس العائل GNP.

دعنا نقول أنك تريد إيجاد كل لاعبي Dodgers. أولاً عليك أن تعد العائل establish parentage (والذي يمكن تحقيقه أما بنداء GU أو GN) وذلك عن طريق إصدار ما يلي:

GU TEAMS (TEAMNUM = 12).

بعد ذلك على أن تجري دورة خلال:

GNP PLAYERS

بعد استرجاعك آخر لاعبي Dodgers وهو (Smith) فإن المسار التالي خلال الدورة ينتج عنه رمز حالة محدد أنه لا توجد حدوثات أخرى من هذا النوع للجزء داخل العائل (أي لا يوجد لاعبين آخرين في Dodgers).

فإذا أردت أن تجد كل حدوثات الأجزاء التابعة لكل الأنواع تحت جذر Dodgers فعليك أن تعد مرة أخرى عائل عند Dodgers ثم تجري دورة خلال

GNP

في هذه المرة في ترتيب تتابعي سابق الترتيب تستخلص كل حدوثات الأجزاء التابعة تحت Dodgers من Doe الى الضربة رقم B13 ثم في المسار التالي خلال الدورة يعيد النظام رمز الحالة محدداً أنك وصلت إلى نهاية حدود العائل.

الإضافة والحذف والتجديد:

الإضافة: عندما يراد إضافة حدوث جزء جديد فيجب أن يضعه البرنامج في منطقة مدخلات ومخرجات. وبعد ذلك يجب إصدار أمر الإضافة Insert من DL/I مع ترك أدنى مستوى SSA دون تأهيل. فإذا أردت على سبيل المثال إدخال لاعب جديد تحت Dodgers فعليك أن تضع حدوث الجزء الجديد في منطقة مدخلات ومخرجات وتصدر ما يلي:

ISRT TEAMS (TEAMNUM = 12)
PLAYERS

فإذا كان الجزء المراد إدخاله عبارة عن جذر فيستخدم النظام الفهرسة المناسبة أو أسلوب التكرار المناسب في إدخاله في الموقع الصحيح لقاعدة البيانات. فإذا لم يكن جذراً فيدخله النظام في السلسلة التوأم المناسبة بإحدى طرق عديدة. فإذا كان نوع الجزء له حقل رئيسي فريد يحفظ على أساسه التتابع فيدخل النظام الجزء في المكان الذي يحتفظ بالتتابع طبقاً لقيمة الحقل الرئيسي للجزء المضاف. فإذا كان نوع الجزء ليس له حقل رئيسي فإنه يوضع في بداية السلسلة التوأم (المشار إليها بالأول first في التكوين) أو في نهاية السلسلة التوأم (last) أو عند النقطة الموجودة عندها مشير الموضع الحالي في السلسلة التوأم حالياً (here) وذلك طبقاً للمواصفات التي تعد في مجموعات التحكم. فإذا كان الجزء به حقل رئيسي

غير فريد فتطبق نفس قواعد (here, last, first) داخل أجزاء قيم equi-key عـلى السلسلة التوأم.

الحذف: الحذف هو ببساطة تحديد حدوث جزء محدد بوضع مشير الوضع الحـالي مشيراً إليـه ثم إصدار نـداء DLET. يمكن أن يوضـع هـذا المشـير بـطريقـة طبيعية باستـدعاء اسـترجاع مثـل: GU أو GN أو GNP. وعلى أيـة حال هنـاك مشكلة ليست مع IMS ولكنها مع أي موقف بيانات مشـتركة يحـاول فيها العـديد من المستفيدين تغيير أحد السجلات في نفس الوقت. وهي تتطلب استخدام رموز وظيفيـة مختلفة بعض الشيء في هـذه الحالـة. سوف نناقش هذه المشكلة فيـما بعد تحت عنـوان التزامن concurrency ويكفي الآن القـول بأنـه هنـاك مجمـوعـة أوامـر متـوازية مـع Get Unique و Get Next و Next Withen Parent تعـرف بـأنها أوامر «Get Hold» وتستخدم عنـدما يحـدث مثل هـذا الموقف. وهي Get Hold Unique (GHU) و Get Hold Next (GHN) و Get Hold Next Within Parent (GHNP) وعـلى هـذا الحـذف حـدوث الجـزء لـرقم الضرب B09 فـإنـك تصدر الأمر:

```
GHU TEAMS (TEAMNUM = 12)
     BATS (BATNUM = B09)
DLET
```

وهناك تعقيد في عملية الحذف وذلك مع العلاقات المنطقية. دعنا ننظر أولاً بدقة أكثر إلى الحذف داخل مثال الهرمية الطبيعية الفردية. إذا حذفنا جـزء المدرب Adams فيجب أن نتوقع أن يحـذف النظام كـذلك أجـزاء خبرتـه وهذا مـا يحدث بالفعل. وليس هناك ما يجعلنا نستطيع أن نحتفظ بأجزاء خبرة أحد المدربين طالما أنه هو نفسه قد حذف. وفي واقع الأمر هذه خاصية للتكوين الهرمي فكل حدوث لجزء باستثناء حدوث الجذر لا بد أن يتصل بحدوث عائل طوال الوقت.

لكن ماذا يحدث إذا ما أشار نـداء DL/I إلى قاعـدة بيانـات منطقيـة تشمل قاعدتين بيانـات طبيعيتين متصلتـين ببعضها بعـلاقة منـطقية. إرجـع إلى الهرميـة المنطقية الموجودة في شكل 6.17 وإلى الهرميتين المتصلتين الموجودتـين في شكل 6.12 المبنى عليها الهرمية المنطقية. إذا حذفنـا حدوث جـزء جذر Dodgers هـل يحذف

كل لاعبي Dodgers من قاعدة البيانات طبقاً لذلك؟ بعد كل شيء أنهم متصلون تحت Dodgers في الهرميات الطبيعية في شكل 6.12 فإنهم موجودين كحدوثات لأنواع أجزاء جذور قائمة بذاتها. والإجابة تعتمد على موقف التطبيق. فقد نريدهم أن يحذفوا أو لا يحذفوا فالنظام عام بدرجة كافية ليسمح بعمل مثل هذا القرار مسبقاً في مجموعات التحكم. ويسمى ذلك في IMS بشفرة القواعد rules coding وهناك مثل هذه القواعد rules للإضافة وللتجديد مثل ما هو موجود للحذف.

التجديد: تسمى عملية التجديد في DL/I بالاحلال Replace ولها رمز وظيفي REPL. وطريقة العمل تقع بين الحذف Delete والإضافة Insert ويوضع الجزء المجدد في منطقة مدخلات ومخرجات (كما في حالة الإضافة). والجزء المراد تجديده يوضع في قاعدة البيانات مع مشير الموضع الحالي وتستخدم أوامر Get Hold (كما في حالة الحذف).

مواضيع إضافية:

رموز الأوامر: يمكن إضافة مؤشرات معينة تسمى رموز الأوامر Command Codes إلى عبارات نداء DL/I. وهي قادرة على توسيع مدى عمل نداءات DL/I وفيما يلي مثالين.

كما سبق أن رأينا فإن نداء SSA المتعدد يسترجع الجزء المسمى في آخر SSA فقط. وهناك مؤشر يعطي إشارة بنداء مسار path call ليسترجع كل (أو أي محددات إضافية) الأجزاء الموجودة في مسار فردي للأجزاء بين أول وآخر SSA محددتين مثال ذلك النداء:

```
GU TEAMS*D (TEAMNUM = 12)
   BATS (BATNUM = B09)
```

سوف يعيد:

12 Dodgers Los Angeles Wilson B09 General

في منطقة المدخلات والمخرجات. «D» مؤشر نداء المسار وهو غير مطلوب في آخر SSA حيث أنه يعاد دائماً.

المؤشر «L» سوف يجد آخر حدوث على سلسلة توأم مثال ذلك.

```
GU TEAMS (TEAMNUM = 12)
   BATS*L
```

تعيد:

```
B13   Acme
```

في منطقة المدخلات والمخرجات.

مشيرات الوضع الحالي المتعددة. سبق أن وضعنا مشير الوضع الحالي كمحافظ لموقع داخل قاعدة البيانات. وهناك مواقف لتطبيقات لا يكون المشير الواحد من هذه المشيرات كافياً. مثال ذلك قد يحتاج البرنامج أن يحفظ تتبع آخر حدوث لكل من أنواع الأجزاء المختلفة التي يتم الاتصال بها. وهناك أسلوبان معروفان بأنهما تجديد مواقع متعددة «multiple positioning» و PCB متعددة «multiple PCB» يسمحان بمدى خيارات في جعل النظام يحتفظ بمشيرات متعددة للوضع الحالي. هذه الخيارات تشمل مثل هذا المشير منفصلاً لكل نوع جزء في الهرمية وحتى مواضع متعددة داخل نوع الجزء.

تسهيلات إعداد التطبيق (ADF):

مفهوم منتج التطبيق: العملية الجارية transaction في نظام IMS هي رسالة يرسلها المستفيد عن طريق نهاية طرفية الى داخل الكمبيوتر والتي تفجر برنامجاً ليبدأ بعض التشغيل وربما يعيد النتائج الى النهاية الطرفية. وبعض العمليات الجارية يصاحبها تشغيل معقد ومؤجل أو إضافة أو حذف جزء أو تجديد بيانات الجزء. وبالتأكيد منطق البرنامج وشفرته اللازمتين لتنفيذ هذه العمليات البسيطة يختلفان قليلاً عن مثل هذا الطلب لبيانات أخرى وإن كانت مختلفة ولتكوينات هرمية ولعروض شاشة مشمولة معها. ويبدو أنه من المخجل أنه يوجد مبرمجون يكتبون مثل هذه الشفرة من البداية. إعادة اختراع العملية في كل مرة يجب تشغيل برنامج جديد فيها عند كتابة مثل هذه العمليات الجارية.

هذا الموقف شائع الوجود مع أي قاعدة بيانات (وبصفة عامة لأي بيئة تشغيل ملفات على مستوى كبير). أحد الاستجابات لهذا هو تطوير منتجات

التطبيقات application generators. ويسمح منتج التطبيق للمبرمج بأن ينفذ عرضاً بسيطاً ويضيف ويحذف ويجدد وظائف للتطبيق في مستوى أعلى وأبسط عن المستوى المقدم بلغة الكوبل المعتادة أو بأي لغة قياسية أخرى. ونمطياً يحتوي منتج التطبيقات على مقاطع نمطية لتنفيذ هذه الوظائف البسيطة.

ويضع المنتج وصفاً لقاعدة البيانات المشمولة ووصفاً لطبيعة المخرجات المطلوبة (مثل تخطيطات الشاشة) لتطبيق معين داخل هذه المقاطع النمطية وينتج شفرة لتحقيق الأعمال المطلوبة.

بالنسبة لتطبيقات DL/I قدمت شركة IBM تسهيلات لإعداد البرامج Application Development Facility (ADF) وذلك لبيئة IMS/DC ونظام إدارة تطوير Development Management System (DMS) شبيه وظيفياً له وذلك لبيئة CICS.

تطوير التطبيقات باستخدام ADF: تطوير التطبيقات باستخدام ADF هو بديل لكتابة البرامج بالكوبل أو بلغة PL/I أو بلغة المجمع التي تتصل بقواعد بيانات IMS. ومن المهم عند هذه النقطة فهم أن ADF ليس لديه أي شيء لعمله مع تصميم أو انتاج قواعد بيانات IMS. ومثل برامج الكوبل الموجود بها نداءات DL/I داخلية فإن ADF عبارة عن وسيلة اتصال موجودة في قواعد بيانات DL/I وتقوم بعرض ومعالجة البيانات فيها.

يحتوي ADF على مقاطع تشغيل نمطية لمعالجة قاعدة البيانات. يقوم مطور التطبيق بتفصيلها خصيصاً لتطبيق معين وتجري عملية التفصيل بتقديم مواصفات وصفية للنظام تسمى بالقواعد rules. وتصف القواعد تكوين قواعد البيانات المراد الاتصال بها وطبيعة العمليات الجارية واعتبارات الأمن التي يمكن عن طريقها أن يؤدي المستفيدون العمليات المحددة لهم على البيانات الخاصة بهم. ووظائف التنقيح والمراجعة المحاسبية التي تختبر للتأكد من أن البيانات التي يتم إدخالها في قاعدة البيانات تقع داخل قيم مدى محددة ورسائل الخطأ لمواقف معينة. وتحدد القواعد كأسطر في جداول وعلى هذا فيسمى ADF بأنه نظام تقوده الجداول «table - drive».

بالإضافة للعمليات الأكثر تعقيداً يستطيع المستفيدون تنفيذ شفرة كوبل أو

PL/I أو مجمع في حالة تشغيل خاص اثناء العمل تحت مظلة ADF.

استخدام ADF: شكل 6.24 هو أول تخيل يراه المستفيد من ADF في عرض الشاشة (بعد تحديد إشارة معينة على الشاشة) هذه القائمة الأساسية ينتجها النظام وهي شاشة نمطية تقدم عدة خيارات للمستفيد. بالنسبة للاتصال بقاعدة البيانات يختار المستفيد «D» محدداً أنه يريد تنفيذ عملية جارية من نظام IMS. بعد ذلك يختار المستفيد حالة العملية الجارية. حالات العمليات الجارية على شاشة القائمة الرئيسية تمثل الأربع وظائف البسيطة التي سبق أن ذكرناها الآن. وسبب وجود ست حالات مسرودة هو أنه يحدث تمييز بين حذف جزء جذر وإضافته («delete» و «insert») وإزالة جزء غير جذري وإضافته («remove» و «add»).

اصطلاح الاسترجاع retrieve يحدد عرضاً بسيطاً للبيانات بينما يعني تجديد update عرضاً ويسمح بالتغييرات.

```
                              P R I M A R Y   M E N U
OPTION:        TRANSACTION MODE:        IDENTIFIER:
               KEY:

   OPTIONS                                      TRANSACTION MODES
A - PROJECT MESSAGE SENDING                        1 - DELETE
B - PROJECT MESSAGE DISPLAY                        2 - INITIATE
C - SESSION TERMINATION                            3 - REMOVE
D - TRANSACTION SELECTION                          4 - ADD
F - PROJECT / GROUP SWITCH                         5 - UPDATE
H - USER MESSAGE SENDING                           6 - RETRIEVE
I - USER MESSAGE DISPLAY

                                    FOR OPTION  - IDENTIFIER IS
                                         D      - TRANSACTION ID
                                         F      - PROJECT/GROUP
                                    A,B,C,H,I   - (NOT USED)
```

شكل 6.24

قائمة ADF الابتدائية (أعيدت طباعتها بإذن من IMSADF GIM (GB21 - 9869 - 2) من شركة IBM)

وبتحديد نوع العملية المراد تنفيذها ينقل النظام على ذلك المستفيد الى

شاشة اختيار الخيارات الثانوية Secondary option selection (شكل 6.25).

```
        S E C O N D A R Y   O P T I O N   S E L E C T I O N   PAGE:
                                                              LAST

INITIATE  SELECT:       KEY:
ACTION:                 SELECT 'C' TO RETURN TO PRIMARY MENU
P3 - PAY/PERS YTD
P5 - PAY/PERS YTD
P7 - PAY/PERS SAVINGS BOND
P8 - PAY/PERS MISC.LEAVE
PA - PAY/PERS PERS+EDUC
PB - PAY/PERS POSITION
PC - PAY/PERS APPOINTMENT
PD - PAY/PERS SEPARATION
C2 - CLASS/COMP GRADE
C4 - CLASS/COMP POSN+HOLDER
T3 - TIME+ATTEND ACCT CHG
T4 - TIME+ATTEND LEAVE
M2 - PAYROLL PROCESS
D2 - DEPT/PAYR PROCESS
A1 - PAY CHK RECON
EA - EMPLOYEE APPOINTMENT
```

شكل 6.25

قائمة خيارات ADF ثانوية (أعيدت طباعتها بإذن من
IMSADF GIM (GB21 - 9869 - 2) من شركة IBM

باستخدام هذه الشاشة يختار المستفيد العملية الجارية التي يرغب في تنفيذها من خلال العمليات الجارية المسموح له بتنفيذها. وتشمل العمليات الجارية توجيهات لكل مما يفعله المستفيد ولنوع الجزء من قاعدة البيانات التي يمكنه استخدامه.

ويبين شكل 6.26 شاشة اختيار المفتاح الرئيسي Primary Key Selection Screen وهي الشاشة التالية التي يراها المستفيد. وهي تحدد الحقول الرئيسية لأنواع الأجزاء التي يمكن الاتصال بها بواسطة العلمية الجارية المطلوبة وتسمح للمستفيد بإدخال قيماً لهذه الحقول الرئيسية. وهذه تذكر للنظام أي حدوث occurence لنوع النظام يريد أن يراه أو يعمل عليه المستفيد.

```
                    PRPS PAYROLL/PERSONNEL APPLICATIONS
                       PRIMARY KEY SELECTION SCREEN
INITIATE     TRANSACTION:  EMPLOYEE APPOINTMENT
OPTION:     TRX:   2EA   KEY:

*** ENTER THE FOLLOWING KEY INFORMATION ***
  EMPL SOC-SEC NUMBER-----------
  PAY-PERS KEY PERSONNEL--------
  PAY-PERS KEY EDUCATION--------
  PAY-PERS KEY POSITION---------
  PAY-PERS KEY APPOINT----------
```

شكل 6.26

قائمة ADF لاختيار المفتاح (الابتدائي) الرئيسي (أعيدت طباعتها بإذن من

IMSADF GIM (GB21 - 9869 - 2)

وأخيراً يعرض حدوث الجزء المطلوب (6.27). إذا وجد حدوث الجزء بالفعل فإن العرض يبين قيم الحقول الفعلية ومعها أسماء الحقول. فإذا لم يحدث ذلك فإنه يبين أسماء الحقول (في الأعداد لعملية إضافة على سبيل المثال). وأول شيء في هذا العرض النهائي يمكن أن يكون ترتيباً تقليدياً للنظام أو يمكن أن يحدد بواسطة مطور التطبيق بطريقة سهلة جداً.

```
            PRPS PAYROLL/PERSONNEL APPLICATIONS

             EMPLOYEE APPOINTMENT (PAGE 1 OF 2)
INITIATE       TRANSACTION:  EMPLOYEE APPOINTMENT
OPTION:       TRX:  2EA  KEY:   548586513B5B5B5B5
*** ENTER DATA FOR UPDATE ***
SOCIAL SECURITY NUMBER: ____________
POSITION NUMBER: _______
TITLE CODE:  GRADE:__  STEP:__  SALARY:     .00  HOW-PAID: _______
EMPLOYEE NAME LAST:__________ FIRST:__________ MI:__  JRSR: _____
PRIOR SERVICE:  0      YEAR LAST WORKED: _____
PROBATION CODE:___       DATE PROBATION END:_____  FLAS:_____
TYPE APPOINTMENT:_____   FULLPART TIME: ___  HIRE DATE:____  MAIN:____
PAY-PERS KEYS: :  PERSONNEL(P9):  B5  EDUCATION(PA)  :  B5
                  POSITION (PB):  B5  APPOINTMENT(PC):  B5
  NEXT SCREEN:ENTER,OR STOP PROCESS:E,OR RETURN TO BEGINNING:R  1
```

شكل 6.27

عرض ADF للبيانات (أعيدت طباعتها بإذن من

IMSADF GIM (GB21 - 9869 - 2) من شكرة IBM

مراجع: References

IBM Corporation, IMS Application Development Facility (IMSADF) General Information Manual IBM From No. GB21-9869, 1980.

IBM Corporation, DL/I Functions for Application Design, IBM Form No. SR20-4680, 1981.

IBM Corporation, IMS/VS Version 1 Application Programming Reference Manual, IBM Form No. SH20-9026, 1978.

IBM Corporation, IMS/VS Version 1 Data Base Administration Guide, IBM Form No. SH20-9025, 1980.

Kapp, D., and Leben, J.F., IMS Programming Techniques, Van Nostrand Reinhold, New York, 1978.

Walsh, M. E., Information Management Systems/ Virtual Storage, Reston Publishing, Reston, VA, 1979.

اسئلة وتمارين: Questions and Exercises

١ - صف الهرمية. صف مثالاً لهرمية في إحدى الشركات.

٢ - صف الشكل الأساسي لهرمية IMS. ماذا تمثله العقد؟ كيف تعامل المصاحبات المتعددة افتراضياً؟ كيف نقدم لتقليل تكرار البيانات؟

٣ - كيف نعدل من هرمية فرق البيسبول Baseball Teams الموجودة في شكل 6.4 لتشمل معلومات عن كل حاملي تذاكر كل موسم؟

٤ - وضح بعناية الفرق بين نوع الجزء وحدوث الجزء.

٥ - قارن بين مميزات وعيوب التخزين المبني على المشيرات والتخزين التتابعي للبيانات في هرميات IMS.

٦ - في هرمية فرق البيسبول الموجودة في شكل 6.4 هل الضروري أن يوجد حقل لرقم الفريق في جزء PLAYERS لماذا ولم لا؟

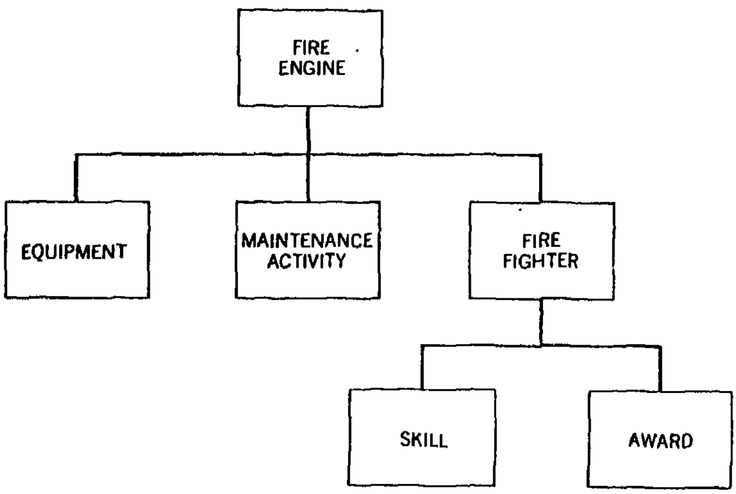
٧ ـ أعد رسوم حدث Dodgers من هرمية فريق البيسبول الموجودة في شكل 6.7 ليعكس الرسم ما يلي (ملاحظة: يظل اللاعبون والمدربون مرتبون طبقاً للحروف الأبجدية).

أ ـ إضافة المدرب Doogan وهو مدرب لديه خبرة مدرسة عملياً وكلية.

ب ـ إضافة اللاعب Evans.

جـ ـ إضافة اللاعب Turner.

٨ ـ صف معنى الهرمية المبينة بالنسبة إلى بيئة العالم الواقعي الذي تمثله (ما هي طبيعة البيانات التي يمكن تخزينها فيها وما هي أنواع التطبيقات التي يمكن أن تستخدمها)؟



٩ ـ صف بكلمات من عندك مميزات العلاقة المنطقية في تقليل التكرار وترويج سلامة البيانات وما إلى ذلك.

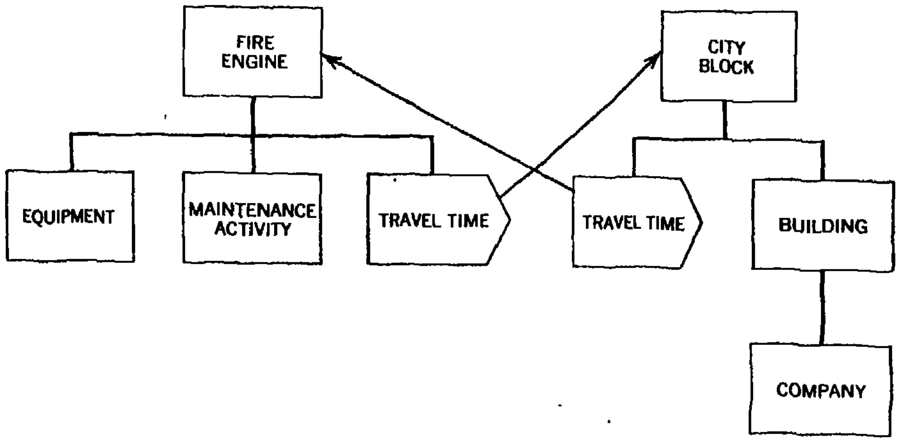
١٠ ـ لماذا تكون هرميات Teams و Players الموجودة في شكل 6.9 حلاً غير كاملاً لتكرار نوعي أجزاء PLAYER في هرميات الشكل 6.8؟

١١ ـ ما مميزات العلاقة الغير منطقية غير الموجهة الموجودة في شكل 6.10؟

١٢ ـ عرف وصف الغرض من الجزء التابع المنطقي. ما بيانات التقاطع ولماذا تنتمي الى التابع المنطقي؟

١٣ ـكيف يمكن استخدام العلاقات المنطقية في تمثيل علاقات العديد للعديد في قواعد بيانات نظام IMS؟

١٤ ـصف معنى قاعدة البيانات الموضحة بالنسبة إلى بيئة العالم الواقعي الذي تمثله (ما هي طبيعة البيانات التي يمكن أن تخزن فيها وما هو نوع التطبيقات التي يمكن أن تستخدمها؟)



١٥ ـ ما الاختلاف بين قاعدة بيانات IMS المنطقية والأخرى الطبيعية؟ وما مزايا عمل هذا التمييز؟

١٦ ـ ما طريقة اتصال IMS (IMS Access Method)؟ صف شكل 6.19 في كلمات من عندك.

١٧ ـ ما الفهارس الثانوية وماذا تحققه هذه الفهارس؟ إذا كان هناك اتصال مباشر بالفعل بجزء الجذر ما الغرض من وجود فهرس ثانوي مبني على حقل في جزء الجذر؟

١٨ ـ ما مجموعة التحكم؟ وما نوع المعلومات التي تحتويها مجموعة التحكم؟

١٩ ـ ما عبارات معالجة البيانات الرئيسية في نظام IMS؟

وهل تغطي هذه العبارات كل أنواع وظائف معالجة البيانات التي قد تحتاجها؟

٢٠ ـ ارجع إلى سجل قاعدة بيانات IMS الموجودة في شكل 6.23 (حدوث للهرمية الموضح في شكل 6.22) وحدد نتائج كل من استدعاءات IMS التالية:

a. GU TEAMS (TEAMNUM = 12)
PLAYERS (PLAYNUM = 5410)
b. GU TEAMS (TEAMNUM = 12)
BATS (BATNUM = B09)
c. GU TEAMS (TEAMNUM = 12)
BATS (MANUF = Acme)
d. GU TEAMS (TEAMNUM = 12)
COACHES (COACHNAM = Adams)
WORKEXP (EXPYRS = 10)
e. GU TEAMS (TEAMNUM = 12)
COACHES (COACHNAM = Adams)
WORKEXP (EXPYRS = 13)

٢١ ـ ارجع إلى سجل قاعدة بيانات IMS الموجودة في شكل 6.23 والذي يكون حدوثاً للهرمية المبينة في شكل 6.22.

أ ـ اكتب نداء DL/I لإيجاد عدد السنوات التي كان فيها Adams مدرباً بخبرة كلية College Coach في Dodgers.

ب ـ اكتب نداء DL/I لايجاد ما إذا كان اللاعب المسمى Murray موجوداً في فريق Dodgers.

٢٢ ـ ارجع الى سجل قاعدة بيانات IMS الموجودة في شكل 6.23 (لحدوث الهرمية الموجودة في شكل 6.22) وحدد نتائج كل من نداءات IMS التالية. افرض في كل حالة أن:

GU TEAMS (TEAMNUM = 12)

تم تنفيذها تكون محددات الدورة لبرامج كوبل أو DL/I.

أ ـ بداية الدورة:

GN PLAYERS

نهاية الدورة

ب ـ بداية الدورة

GNP PLAYERS

نهاية الدورة

جـ ـ بداية الدورة

GNP BATS (MANUF = Acme)

نهاية الدورة

د ـ GN COACHES

بداية الدورة

GNP WORKEXP

نهاية الدورة.

٢٣ ـ ارجع الى سجل قاعدة البيانات IMS الموجود في شكل 6.23 والذي يكون حدوثاً للهرمية المبينة في شكل 6.22. اكتب استدعاءات DL/I بدورات إذا كان ضرورياً (انظر النظام الموجود في التمرين السابق) للحصول على:

أ ـ كل مدربي Dodgers.

ب ـ كل خبرة التدريب لـ Adams الموجود في Dodgers.

جـ ـ كل خبرة التدريب لكل من مدربي Dodgers.

٢٤ ـ ارجع الى سجل قاعدة البيانات IMS الموجود في شكل 6.23 والذي يكون حدوثاً للهرمية المبينة في شكل 6.22. اكتب نداءات DL/I بدورات إذا كان ضرورياً لعمل:

أ ـ ادخال مدرب جديد اسمه Turner في آخر سجل قاعدة بيانات Dodgers.

ب ـ حذف كل خبرة المدرب Adams.

جـ ـ تجديد عمر اللاعب Jones إلى 26.

٢٥ ـ ما منتج التطبيقات؟ وما مميزات استخدامه؟ وهل له عيوب؟

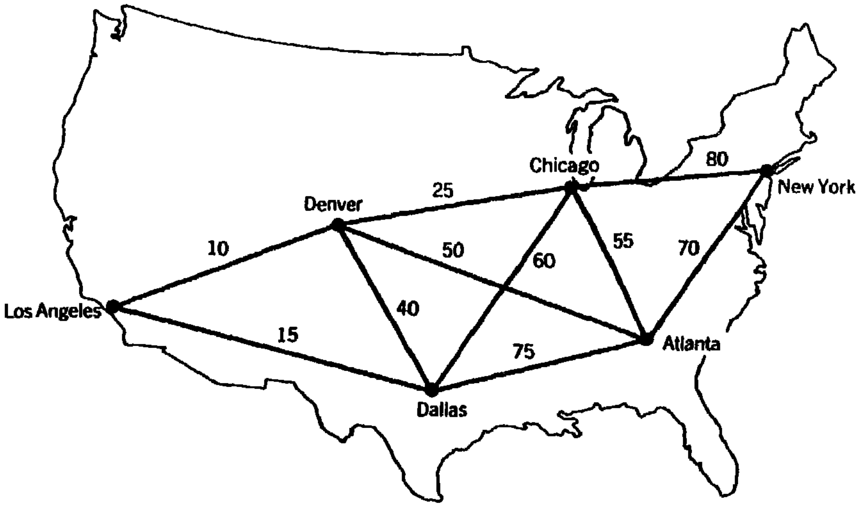
الفصل السابع
المنهج الشبكي لقاعدة البيانات
The Network Approach to Database

مقدمة : Introduction

هناك منهج آخر لنظم ادارة قواعد البيانات يعتمد على تكوينات تسمى شبكات. عبر العقدين الميلاديين السابقين طورت العديد من نظم إدارة قواعد البيانات وهي تنفيذات إلى حد كبير أو إلى حد أقل لشبكات موجهة لمواصفات نظم إدارة قواعد بيانات طورها كل من مؤتمر لغات نظم البيانات Conference on Data System Languages (CODASYL) ومجموعة أنشطة قواعد البيانات Data Base Task Group (DBTG) ولجنة لغة وصف البيانات Data Description Language committee (DDLC). ومؤتمر لغات نظم البيانات هو منظمة مكونة من متطوعين يمثلون مصنعي أجهزة الكمبيوتر والمستفيدين في الصناعة والحكومة الفيدرالية لكل من كندا وأمريكا.

وعمل مؤتمر CODASYL على عمل مواصفات منظمة لنظم إدارة قواعد البيانات بدأ في نهاية الستينيات الميلادية. وقد انتجت تقارير بمواصفات CODASYL من وقت لآخر وبعض هذه التقارير الملحوظة ظهرت في 1971 م و 1973 م و 1978 م و 1982 م. ونظم إدارة قواعد بيانات CODASYL المختلفة المتاحة تجارياً مبنية على بعض النسب المئوية للمواصفات الموجودة في أحد تقارير CODASYL أو في أكثر من تقرير من هذه التقارير.

وللحفاظ على محتويات هذا الكتاب في تقديم فهماً دقيقاً وواضحاً للمفاهيم الأساسية لقاعدة البيانات فإننا نناقش منهج CODASYL لنظم إدارة قواعد البيانات كموضوع عام. ولن نميز بين صيغ مواصفات CODASYL المختلفة أو التنفيذات التجارية المختلفة وما سنفعله هو وصف المعالم الرئيسية والأهداف العامة لمنهج CODASYL.



شكل 7.1 شبكة الطرق السريعة

الشبكات : Networks

لكي نبدأ دعنا ننظر إلى ما تعنيه كلمة شبكة network. فالشبكة هي نوع عام جداً من التكوين: ببساطة فهي مجموعة من العقد nodes متصلة مع بعضها بطريقة معينة. وسوف نشير بهدف الإقناع إلى الاتصال بين عقدتين بأنه حافة edge. هل مفهوم الشجرة يناسب هذا التعريف؟ نعم إنه يفعل ذلك مع بعض التحفظات ففي الواقع الشجرة هي نوع مقيد من أنواع الشبكات.

وعملياً نتعامل مع شبكات بصفة متكررة في حياتنا اليومية. يبين شكل 7.1 إحدى مثل هذه الشبكات الشائعة الاستخدام: شبكة الطرق السريعة التي تصل بين المدن بعضها البعض. ومثال آخر هو شبكة الهاتف. كما أننا نتكلم كذلك عن شبكة أصدقاء يلتقي فيها الأفراد بعضهم ببعض.

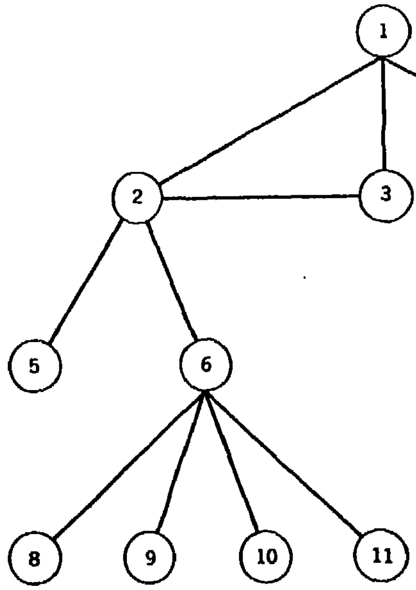
وتعريف المصطلحات التالية الآن يساعد فيما بعد. تكون العقدتان متجاورتان adjacent إذا كانتا متصلتين مع بعضهما بواسطة حافة. وعلى هذا فشيكاجو ودنفر متجاورتان في شكل 7.1 حيث أنهما متصلتان ببعضها بالطريق السريع رقم 25. تكون الحافة متماسه incident لنقطة إذا مستها. فالطريق السريع 15 متماس مع دالاس. درجة degree العقدة node هو عدد الحافات المتماسة لها. المسار path هو سلسلة حافات متصلة على التوالي مثل الطرق السريعة 10 و 25 و 80 والتي تصل لوس انجيلوس بنيويورك. والدورة Cycle هي

مسار يبدأ وينتهي عند نفس العقدة. المسار الخاص بالطرق السريعة 25 و 55 و 75 و 40 هو دورة لأنه يبدأ وينتهي في دنفر.

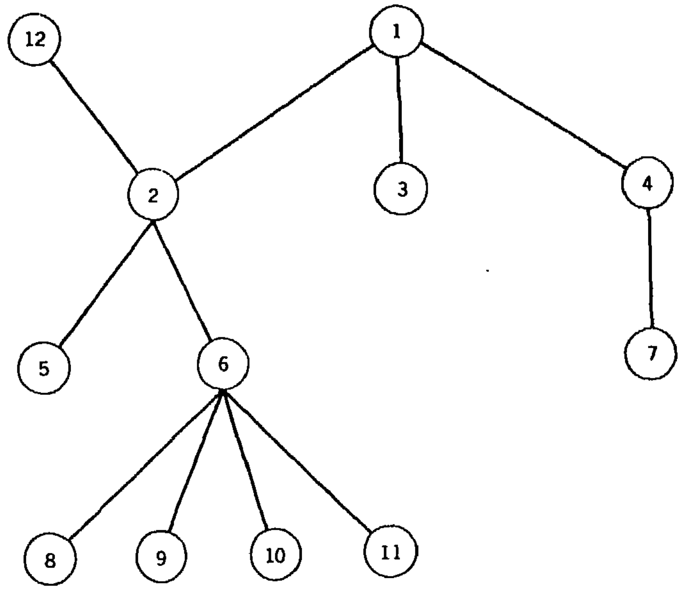
هذه الاختلافات بين الأشجار والشبكات الأكثر عمومية مهم بالنسبة للمناقشة التي تلي. فهناك زوج من الطرق للتمييز بين الشجرة والشبكة الأكثر عمومية. وبينما يمكن لمثل هذه المناقشة أن تصبح معقدة بسهولة فإننا نحاول أن نجعلها أو نحافظ على أن تكون بسيطة بقدر الإمكان.

في العادة الشجرة هي شبكة ليس لها دورات. ارجع الى شكل 6.1 وأقنع نفسك أنه لكل عقدة في الشبكة والتي نسميها شجرة لا توجد أي طريقة لبدء مسار عند عقدة وافترض سلسلة من الحافات والعودة الى عقدة البداية دون الحاجة إلى العودة للخلف إلى إحدى الحافات على الأقل مرة واحدة على الأقل. ضاهي ذلك مع تكوين شكل 7.2 والذي يبين دورة تصل النقاط 1 و 2 و 3.

لكنه من الممكن كذلك أن توجد شبكة ليست شجرة وليس بها دورات أيضاً. فعند الكلام عن الأشجار في الفصل السادس قمنا بعمل الجذر كعقدة في قمة التكوين وعملنا دائماً لأسفل من عند الجذر. وتأثير ذلك هو أننا حددنا اتجاه direction على الأفرع أو الحافات. وهو لأسفل (والذي يناظر كذلك لعلاقة من



شكل 7.2 ، شبكة بها دورة



شكل 7.3
شبكة بدون دورات

واحد إلى العديد في نفس الاتجاه DL/I). وبمجرد تحديد مثل هذا الاتجاه لأسفل على حواف الشجرة فهناك سمة أخرى للشجرة وهي أن النقطة لا يكون لها أكثر من عائل واحد. شكل 7.3 هو نسخة من شكل 6.1 مع إضافة عقدة جديدة (رقم 12). وهو يمثل شبكة (وليست شجرة). لا يوجد لها أي دورات. إذا اعتبرت كل الحواف أنها تشير إلى أسفل الصفحة فمن الواضح أن العقدة 2 يكون لها عائلين مما يجعل هذا التكوين لا يصلح أن يكون شجرة لكنه لا يزال يمثل شبكة صحيحة.

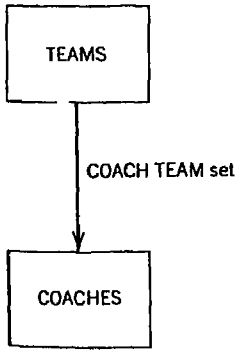
التكوين الطبيعي لمؤتمر CODASYL:

CODASYL Physical Structure

لوصف كيفية تخزين البيانات في صورة شبكية فإننا نستخدم نفس مثال البيسبول الذي سبق استخدامه في مناقشة نظام IMS.

مفهوم الفئة :

مقدمة : مجموعة بناء CODASYL الأساسية للشبكة هي الفئة Set. وتحتوي الفئة على نوعين سجلات في علاقة من واحد إلى العديد لكل منهما. يبين شكل 7.4 فئة COACH TEAM فأي فريق له عدد من المدربين يعمل له.

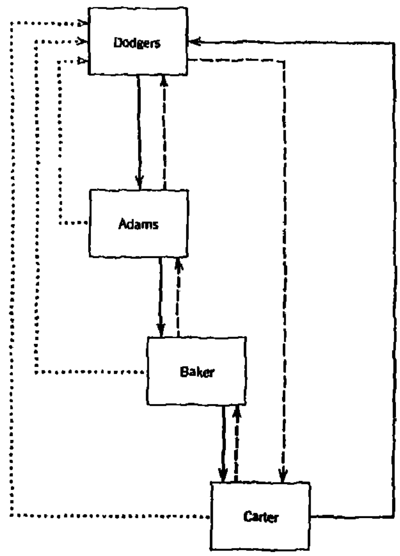


شكل 7.4
فئة CODASYL

وسوف نتبع الاصطلاحات الموجودة عادة في مراجع CODASYL بأن الاتصال بين نوعي السجلات للفئة يوضح كسهم له رأس واحدة (بدلاً من سهم له رأسين) بالرغم من أنه يمثل علاقة من واحد إلى العديد. وعلى عكس وصف نظام IMS سوف نترك رؤوس الأسهم الآن لأننا سوف نرى حالاً أن اتجاه علاقة من واحد إلى العديد لا يكون لأسفل دائماً في الرسومات كما هو في IMS.

نوع السجل الموجود في جانب واحد من علاقة من واحد الى العديد TEAMS يسمى نوع السجل المالك Owner record type بينما السجل الموجود في عدة جوانب COACHES فيسمى بنوع السجل العضو member record type.

تخزين الفئات: كما سبق أن ذكرنا تماماً بالنسبة لحدوثات السجلات وحدوثات الهرميات فيمكننا أن نتكلم هنا عن حدوثات الفئات. يحتوي حدوث الفئة على حدوث واحد لنوع سجل مالك للفئة وكل حدوثات أنواع السجلات الأعضاء المصاحبة له. ويبين شكل 7.5 حدوث فئة Dodgers. COACHE TEAM له ثلاث مدربين هم Adams و Baker و Carter. كما يوضح شكل 7.5 كذلك الطريقة المعتادة التي يخزن بها CODASYL حدوث الفئة: بالمشيرات المشير الرئيسي (وهو الوحيد المطلوب) هو المشير التالي هو مبين كخطوط متصلة في شكل 7.5. وبالنسبة للمشيرات التالية فإن حدوث السجل المالك (Dodgers) يشير إلى أول حدوث لسجل عضو (Adams) ويشير حدوث هذا السجل العضو الى السجل التالي وهكذا حتى حدوث آخر سجل عضو (Carter) في الفئة والذي يعيد للإشارة الى حدوث السجل المالك.

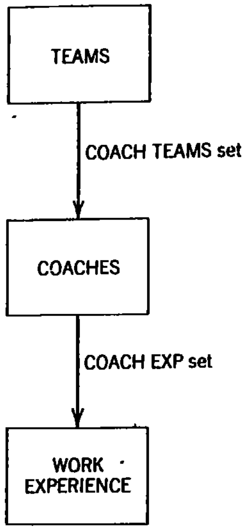


شكل 7.5
حدوث فئة COACH TEAM

لاحظ أنه باستثناء آخر مشير تالي يذهب من آخر حدوث لسجل عضو الى الخلف الى حدوث السجل المالك فإن سلسلة المشيرات التالية تشبه ترتيب المشير المعتمد ـ التوأم المستخدم في IMS. وبالمناسبة فقد يكون مغرياً لك أن تنظر إلى سلسلة المشيرات وتحتج على المشيرات المتتالية أي الدورة Cycle وقد يكون هذا صحيحاً، فهي تشكل دورة فعلاً. تذكر فقط أننا لدينا هنا دورة لحدوثات سجلات record occurence داخل حدوث فئة بينما من الضروري في مناقشة الدورات السابقة في هذا الفصل أنها تتسق تجاه دورات لأنواع سجلات record types وهذه مناقشة سنعود إليها فيما بعد.

هناك نوعان بديلان موضحان في شكل 7.5 ويستخدمان في الواقع العملي مع ترتيبات شبكات أكثر تعقيداً. وتشغيل هذه الشبكات المشير السابق prior (الموضح كخطوط غير متصلة) يظهر بنفس الطريقة التي يظهر بها المشير التالي إلا أنه يذهب في الاتجاه العكسي. المشير المالك owner (الموضح كخط من النقاط) يشير من كل حدوث سجل عضو إلى حدوث السجل المالك للفئة.

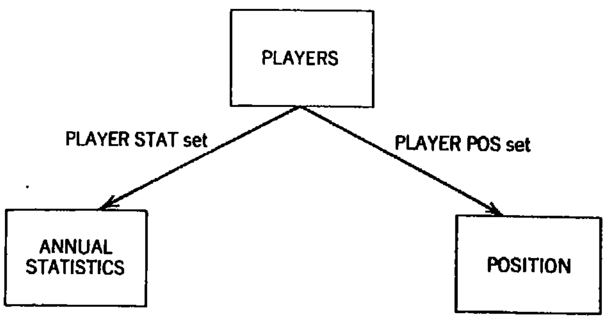
من الفئات الى الشبكات: يمكن أن تدمج الفئات مع بعضها. لتكوين شبكات لها أشكال وأحجام واسعة المدى. يبين شكل 7.6 فئتين. وبالنسبة لعلاقات من واحد إلى العديد المعتادة فإن فئة COACH TEAMS تسجل المدربين للفريق بينما فئة COACH EXP تسجل قطع أعمال خبرة مدرب معين. لاحظ إن نوع سجل COACHES هو نوع سجل عضو لفئة COACH



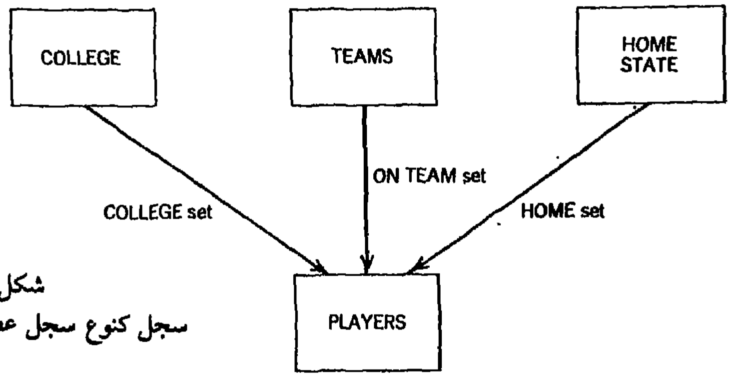
شكل 7.6
سجل كنوع سجل عضو لفئة واحدة ونوع سجل المالك لفئة أخرى.

TEAMS كما إنه نوع سجل مالك في نفس الوقت لفئة COACH EXP ويبين شكل 7.7 فئتين لهما نفس نوع السجل المالك: PLAYERS.

لاحظ أن الشبكات الموضحة في شكل 7.4 وشكل 7.6 وشكل 7.7 (يمكن النظر إلى شكل 7.4 على أنه شبكة تحتوي على فئة واحدة) تظهر على السطح لتكون غير مميزة عن الاشجارة. انظر الآن إلى شكل 7.8 والذي يبين ثلاث فئات. تحدد فئة COLLEGE كل اللاعبين الذين أتوا من كلية معينة كما تبين فئة ON TEAM كل اللاعبين الموجودين في فريق وتبين فئة HOME كل اللاعبين المولودين في ولاية معينة. وفي هذا التكوين كل الثلاث فئات لها نفس نوع السجل العضو PLAYERS. وباستخدام نفس اصطلاحات IMS يمكننا القول أن سجل PLAYERS له ثلاثة أنواع سجلات عائلة parent (والذي لا يمكن حدوثه بالطبع في IMS). وعلى هذا فهذا التكوين ليس شجرة لكنه شبكة أكثر عمومية.



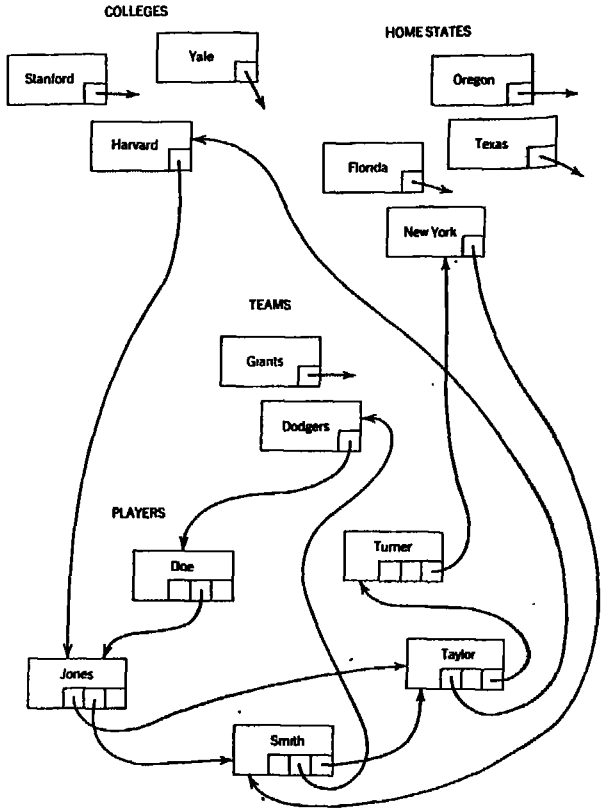
شكل 7.7
سجل كنوع سجل مالك لفئتين مختلفتين



شكل 7.8
سجل كنوع سجل عضو لثلاث فئات مختلفة

يبين شكل 7.9 ترتيب المشيرات لحدوث كل من هذه الفئات الثلاث. ويسهم حدوث السجل في سلسلة واحدة لحدوثات سجلات (ويجب أن يكون هناك مكان للمشير التالي في السلسلة) وذلك لكل فئة موجود فيها. ففي تكوين شكل 7.8 يكون سجل PLAYERS عبارة عن نوع سجل عضو لكل من الثلاث فئات وهذا يعني أن اللاعب يمكن أن يبين على أنه خريج إحدى الكليات وأنه يلعب لإحدى الفرق (حالياً أو سبق له ذلك) كما أنه ولد في إحدى الولايات. وفي حدوث PLAYERS في شكل 7.9 فإن فتحات المشير الموجودة على أقصى اليسار هي لفئة COLLEGE والفتحة الوسطى لفئة ON TEAM والفتحة اليمنى لفئة HOME. ويبين الشكل أن Jones و Taylor خريجان من جامعة Havard وإن Doe و Smith و Jones يلعبون لفريق Dodgers وإن Smith و Taylor و Turner ولدوا في New York. فإذا كانت كل البيانات مبينة فيمكننا أن نرى سلسلة الكلية الموجود عليها Doe وسلسلة الولاية الموجود عليها Jones بأنه ولد فيها وما إلى ذلك.

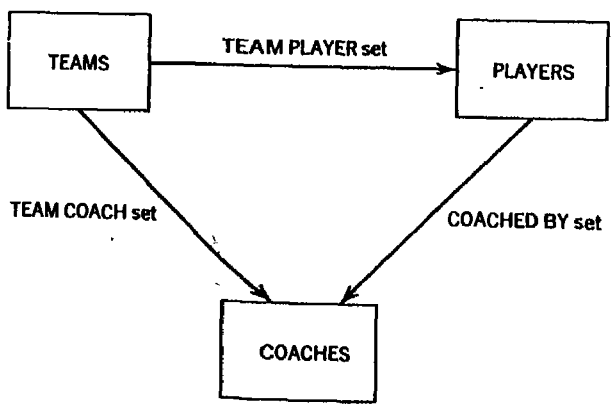
يبين شكل 7.10 تكوين آخر لشبكة ليست شجرة. ويمكنها أن تحتفظ باللاعبين في أحد الفرق في فئة TEAM PLAYER والمدربين في أحد الفرق في فئة TEAM COACH والمدربين الذين يدربون لاعب معين في فئة COACHED BY. أما إذا أردنا أن تحتفظ كذلك بعكس فئة COACHED BY أي باللاعبين المسؤول أحد المدربين عن تدريبهم فإن الإثنين معاً يكونان علاقة من العديد إلى العديد والتي تقود إلى المناقشة التالية عن سجلات الاتصال Juncture records.



شكل 7.9
حدوث كل من الثلاث فئات الموجودة في شكل 7.8 في حدوثات اللاعبين: فتحة المشير اليسرى تكون لسلسلة الكلية وفتحة المشير الوسطى تكون لسلسلة الفريق أما فتحة المشير اليمنى فهي لسلسلة المدينة التي يقطن فيها.

سجلات الاتصال:

حتى الآن لم نقل أي شيء عن تخزين البيانات في علاقات من العديد إلى العديد في شبكات CODASYL. في الواقع (وبطريقة تشبة حالة IMS) فإن تكوينات شبكة CODASYL لا يمكنها أن تمثل علاقات من العديد إلى العديد بصورة مباشرة. يمكننا القول بأنه داخل الفئة الواحدة يمكن أن يصاحب حدوث

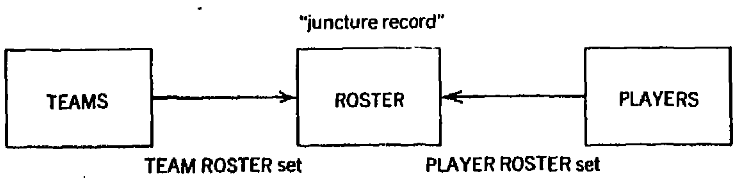


شكل 7.10
تكوين شبكي آخر

سجل عضو حدوث سجل مالك فردي في شكل 7.4. يمكن أن يصاحب المدرب فريق واحد فقط ويمكن ألا يكون هناك بيانات تاريخية في كل الفرق بأن المدرب عمل هناك أثناء فترة عمله كلها، وتعد علاقات من العديد إلى العديد في شبكات CODASYL من خلال تقديم نوع سجل جديد يعرف بأنه سجل اتصال أو اتصال link أو توصيل intersection أو connection. وبدلاً من اتصال فردي من العديد إلى العديد بين نوعي سجلات يمثلان في إحدى الفئات فيكون هناك فئتين. إحدى الفئتين يكون بها أحد نوعي السجلات كنوع سجل مالك وسجل اتصال جديد يعد كنوع سجل عضو لهذا السجل. أما الفئة الأخرى فيكون بها نوع السجل الآخر كنوع سجل مالك ونفس سجل الاتصال الجديد كنوع سجل عضو. وهذا الوضع يشبه في الواقع تكوين العلاقة المنطقية الموجودة في IMS.

يبين شكل 7.11 علاقة من العديد بين TEAMS و PLAYERS على نفس الأساس التاريخي الذي نتعامل به في آخر فصل والممثل في صورة شبكة CODASYL. تؤخذ فئة TEAM ROSTER وفئة PLAYER ROSTER معاً من علاقة من العديد إلى العديد. ويكون هناك حدوث سجل ROSTER بكل لحظة لكل لاعب معين لعب في فريق معين أو العكس. ويوضح شكل 7.12 (والذي يبين نفس العلاقات المبينة في شكل 6.14) حدوثات للفئتين الموصوفتين في شكل 7.11.

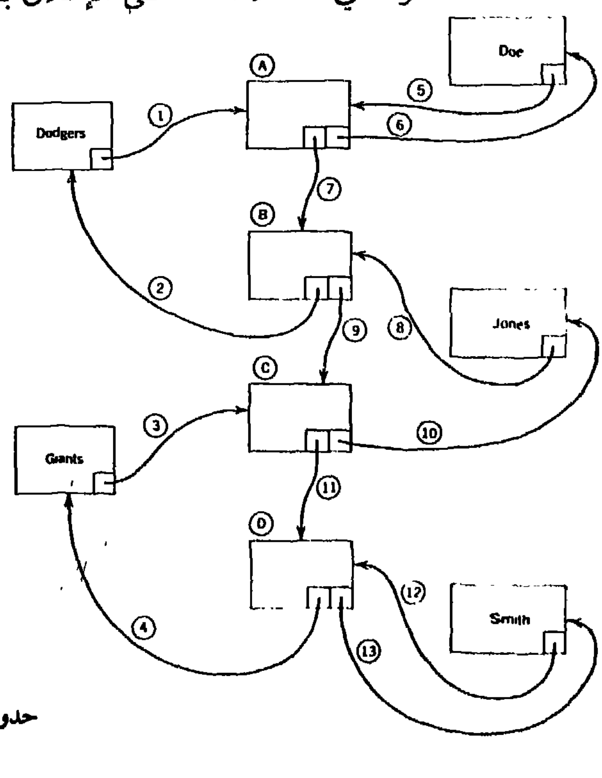
حدوثات سجل ROSTER والمشيرات مسماة للإشارة لها في المناقشة التالية. حدوثات سجل ROSTER المرسومة في منتصف شكل 7.12 كل منها له فتحتين لمشيرين تاليين حيث أن نوع سجل ROSTER هو نوع سجل عضو لكل من الفئتين المشموليتين. في كل حدوث كل مشير موجود في الناحية اليسرى يكون لفئة TEAM ROSTER بينما المشير الموجود على اليمين يكون لفئة PLAYER ROSTER. يكون عند Dodgers كل من Doe و Jones كلاعبين في إحدى الأوقات بينما Giants لديه Jones و Smith. على العكس فإن Doe لعب في



شكل 7.11 علاقة من العديد إلى العديد من CODASYL

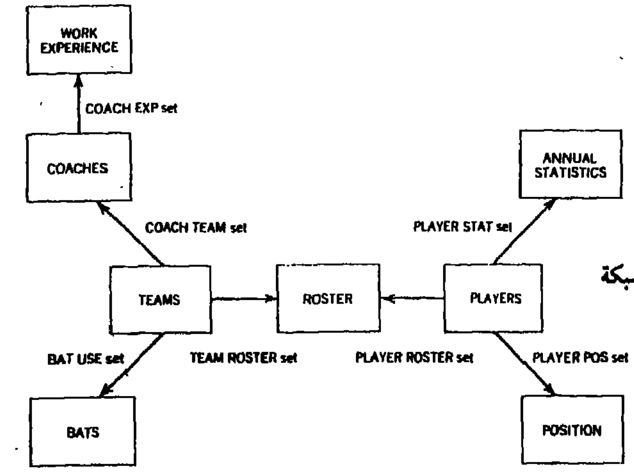
Dodgers فقط و Smith لعب في Giants فقط أما Jones فقد لعب للاثنين.

كما يقدم شكل 7.12 كذلك توضيحاً جيداً للحاجة إلى مشير مالك بديل. تذكر أن كل المشيرات المبينة في شكل 7.12 هي مشيرات تالية. إفرض أنك بدأت عند حدوث Jones وأردت أن تجد كل الفرق التي لعب لها ومعلومات أخرى عن هذه الفرق مخزنة في حدوثات سجل TEAMS. يقود المشير 8 من Jones النظام الى حدوثات Roster B. ويمكن القول عند هذه النقطة أن النظام يحتفظ بمكانه عند B. ويتبع المشير 2 إلى Dodgers منتجاً أحد الفرق التي لعب لها Jones وهذا لأن B كان آخر اللاعبين في سلسلة Dodgers فقط وبالتالي فإنه أشار إلى Dodgers. الحالة الأكثر عمومية والأكثر تعرضاً للمشاكل موضحة بتتبع المشير 9 إلى حدوث سجل الجدول التالي في سلسلة Jones وهو سجل الجدول C roster من أجل الوصول إلى سجل الفرق المصاحب لسجل الجدول C (Giants). يجب أن يتبع النظام المشير 11 ثم المشير 4 (وبصفة عامة من الممكن أن يتبع العديد من المشيرات الأخرى في سلسلة أطول). وفي عمل ذلك يمر النظام خلال حدوث ROSTER D والذي لا علاقة له على الإطلاق باللاعب Jones. ومن ناحية



شكل 7.12

حدوثات للفئتين المبينتين من شكل 7.11



شكل 7.13
مثال الفرق واللاعبين في صورة شبكة

أخرى إذا كانت حدوثات ROSTER لها مشيرات مالكة فيمكن للنظام أن يستخدم من C مشير مالك لإيجاد Giants مباشرة وهو السجل المالك لهذا الحدوث في فئة TEAM ROSTER.

يبين شكل 7.13 (والذي يحدد سجل ROSTER الموجود في شكل 7.11) شبكة اللاعبين والفرق كاملة. تذكر أن بيانات التقاطع سوف تخزن في سجل اتصال. وهذه الشبكة هي مكافيء CODASYL لتكوين IMS المبين في شكل 6.15.

المخططات والمخططات الجزئية:

باتباع الاصطلاحات المتفق عليها لنظم إدارة قواعد البيانات DBMS فإن مواصفات CODASYL تسمح بالتمييز بين البيانات المخزنة فعلياً ومنظرها الذي يستخدمه المبرمج. بالإضافة إلى ذلك فإنها تقترح مستوى مرحلياً للرؤية. وهذا المنظر الذي يوصف على أنه مستوى افتراضي conceptual level يصف قاعدة البيانات كلها ويمكن أن يشمل عدة شبكات منفصلة عن بعضها انفصالاً واقعياً.

وفي CODASYLيسمى وصف البيانات المخزنة فعلياً بالمخطط الداخلي internal schema. ويسمى الوصف عند المستوى الافتراضي بالمخطط المنطقي logical schema وصف البرنامج أو رؤية المبرمج للبيانات يسمى بالمخطط الجزئي subschema. وبالنسبة لأغراض الأمن والوضوح يمكن وصف جزء متصل من شبكة CODASYL في صورة مخطط جزئي يقدم إلى المبرمج لاستخدامه في أحد

التطبيقات. ولا يحتاج المبرمج لأن يهتم ببقية الشبكة الفعلية المخزنة والمعرفة في المخططات.

طرق الاتصال:

بدائل: لقد قدمت الصيغ المختلفة من مواصفات CODASYL وتنفيذاتها المختلفة العديد من الطرق لتخزين حدوثات السجلات والاتصال بها. ومع نظرنا الى البدائل تذكر أن كل نوع سجل في CODASYL بالرغم من أنه قد لا يحتاج أن يعالج مستقلا إلا أنه يعالج مستقلاً عن كل أنواع السجلات الأخرى بالنسبة للاتصال. وسوف نوضح ذلك بالتفصيل الآن:

إحدى تنظيمات CODASYL الشائعة لها حدوثات لنوع سجل تخزن باستخدام طريقة متكررة مبنية على مفاتيحها. وعندما تخزن بهذه الطريقة يقال أن الاتصال عن طريق الحساب via calc . وكما هو معتاد بالنسبة للتنظيمات المتكررة فإن هذه الطريقة تقدم طريقة اتصال سريعة جداً لكن بدون تتابع واقعي للبيانات.

هناك طريقة أخرى لتخزين حدوثات نوع السجل والاتصال بها تعرف بأنها عبر الفئة via set. ونوع السجل المعني يجب أن يكون في نوع سجل العضو من الفئة. ونقطة هذه الطريقة هي تخزين حدوثات السجل العضو لحدوث فئة واقعياً بالقرب من حدوث السجل المالك للفئة. ويجب أن يحدث الاتصال بهذه السجلات الأعضاء عن طريق حدوث السجل المالك للفئة. ولا يمكن استرجاع هذه الحدوثات مستقلة.

هناك طريقة أخرى تسمى بالطريقة المباشرة direct في CODASYL وقد اسقطت من آخر فئة للمواصفات لكن مازالت موجودة في بعض التنفيذات. وفي هذا الترتيب يحدد عنوان نسبي لكل من حدوثات السجل. ويسمى هذا العنون بمفتاح قاعدة البيانات data base key. يمكن للبرنامج أن يميز قيم مفتاح قاعدة البيانات ويحفظها وفي وقت لاحق يشير مباشرة إلى حدوث أحد السجلات بمفتاح قاعدة البيانات الخاص به. وهذه الطريقة هي طريقة مقنعة في انتاج ملفات تتابعية.

وتسمح بعض التنفيذات بتخزين تتابعي مفهرس، كما يسمح البعض الآخر بفهارس ثانوية كبدائل.

تعليقات الاستخدام: تذكر أنه يمكن معالجة كل سجل مستقلاً بالنسبة للاتصال إذا كان هذا له معنى بالنسبة لفئة التطبيق التي تستخدم البيانات. نظرياً كل نوع سجل من الشبكة يمكن أن يخزن عبر الحساب via calc باستخدام طريقة تكرارية. وحدوثات السجل هذه لأنواع سجلات سوف تكون مبعثرة على مدى طبيعي واسع من اسطوانات الأقراص ومجموعات الأقراص إلا أنها تكون مرتبطة ببعضها بطريقة الفئة المعتادة في CODASYL بالمشيرات التالية (على الأقل). وفي هذا الترتيب يمكن أن يحدث اتصال سريع مباشر بأي حدوث لأي نوع سجل. وعلى أية حال فإن الإبحار خلال الشبكة من حدوث إلى حدوث آخر سوف يقل نسبياً بسبب بعثرة خواص التكرار للبيانات.

وطريقة تجنب هذه المشكلة هي تخزين أحد أنواع السجلات بطريقة التكرار وتخزين الأنواع الأخرى المتصلة به وبالتالي الأنواع المتصلة بها عبر الفئة via set. وهذا يسمح بالاتصال المباشر السريع لأحد السجلات. ثم بما أنهم مخزنين واقعياً قريبين من بعض على القرص فيحدث اتصال تتابعي سريع نسبياً للحدوثات المتصلة به في الشبكة. وعلى أية حال عند عمل ذلك فإننا نعطي الاتصال المباشر للسجلات الأخرى (إلا إذا كان مسموحاً باستخدام فهارس ثانوية).

وبالإشارة مرة أخرى الى شكل 7.4 إذا كانت سجلات COACHES مخزنة عبر الفئة بالنسبة إلى سجلات TEAMS وكانت سجلات WORK EXPERIENCE مخزنة عبر الفئة بالنسبة إلى سجلات COACHES فعند ذلك يكون الفريق وسلالاته مخزنين واقعياً بالقرب من بعضها وهذا يكون جيداً للأداء طبقاً لطبيعة التطبيقات التي تستخدم البيانات.

وهذا هو نوع من الإحساس النمطي المشترك لموقف التبادل. ويمكن عمل نفس التعليقات بالنسبة لنظم أخرى بما في ذلك IMS.

السطح البيني للغة CODASYL:
CODASYL Language Interface

مقدمة:

منظمة CODASYL مسؤولة عن حفظ نمطيات للغة الكـوبل ونتيجـة لهذا الاتجاه فإن اتجاه مواصفات قاعدة بيانـات CODASYL كان نـاحية الاتصـال عن طريق برامج الكوبـل بقواعـد البيانـات وبينها تم بنـاء طرق عـديدة من الاتصـال اللغوي في تنفيذات CODASYL المختلفة لقواعد البيانـات فهي تميل أسـاساً لأن تكون وظائف داخلية في برامج الكوبل. وسوف نوضح العبارات المختلفة الموجودة في مثل هذا الاتصال لكننا يجب أن ننظر أولاً إلى عدة أمور مبدئية.

أمور مبدئية:

نوع عضوية الفئة: يمكن أن يوجد حدوث لنوع سجل، والذي يكـون نوع سجل عضو لفئة معينة، وذلـك في قاعـدة بيانـات دون أن يكون في حـدوث هذه الفئـة (دون أن يكون مصـاحباً بحـدوث لنوع سجـل مالـك للفئة). مثـال ذلـك وبالعودة إلى شكل 7.4 يمكن أن يكون للمدرب سجل في قاعدة البيـانات دون أن يكـون في حدوث فئـة COACH TEAM وهذا يعني ببسـاطة أن هـذا المدرب لا يعمل بصفة مؤقتة.

ولنفس السبب فيجب عـلى مصمم قاعـدة بيانـات CODASYL أن يحـدد خواصاً معينة لأنواع السجلات للفئات التي تكـون فيها هـذه السجلات أعضـاء. وتوجه هذه الخواص النظام فيما إذا كانت عبارات معالجة سجـل معين وفئـة معينة قـادرة على أن تؤثـر على سجـلات خاصـة أم لا وإذا كـانت قـادرة فـما هي كيفيـة حـدوث ذلك. وهنـاك بديـلان لنوع التخـزين storage class وثلاثـة بدائـل لنـوع الإزالـة removal class. وأي نوع سجـل يمكن أن يكون لـه أي خليط من نـوع التخزين ونوع الإزالة لأي فئة يكـون عضواً فيهـا. وبصفة عـامة يمكن أن يكـون للسجل مثل هـذا الخليط المختلف لكل فئـة يكون فيهـا نوع سجـل عضو. دعنـا ننظر إلى هذه البدائل الآن كأحد المناطق اللازم تغطيتها قبـل أن ننظر إلى عبـارات معالجة البيانات فيما بعد.

نوع التخزين: نوع تخزين السَجل (ويسمى نوع الإدخـال insersion class

كذلك). يوجه DBMS في اجراءاته عند إدخال حدوث جديد لسجل داخل قاعدة بيانات عندما يعمل هذا السجل كحدوث سجل عضو في الفئة.

إحدى الإمكانيات هي أن يوضح نوع السجل تلقائياً automatically بالنسبة إلى الفئة التي يكون سجلاً عضواً فيها. وبعد ذلك عندما يتم إدخال حدوث هذا السجل في قاعدة البيانات فإن النظام يصله تلقائياً بحدوث هذه الفئة (ويصاحبه ـ أي يوصله ـ بحدوث نوع السجل المالك). بالطبع يجب أن يحدد البرنامج أي حدوث لهذه الفئة هو الذي يصاحب السجل. الخيار الآخر يدوي manual إذا ما وضح نوع سجل عضو بأنه يدوي فعندما يتم إدخال حدوث له في قاعدة البيانات فلا يتصل بحدوث الفئة وبالتالي إذا أصبح ضرورياً فإن ذلك يؤدي بأمر منفصل سوف يتم وصفه فيما بعد.

نوع الإزالة: نوع إزالة السجل (يسمى بنوع الاحتفاظ retension class كذلك) يعطي إذناً للنظام لإطاعة أوامر برنامج التطبيق بنقل حدوثات سجل عضو يصاحبه بالفعل حدوث فئة.

إذا ما وضح نوع سجل والذي يكون نوع سجل عضو لفئة معينة على أن ثابت fixed للفئة فيجب أن يبقى على ذلك في حدوث هذه الفئة التي تم إدخاله فيها أصلاً. ولا يمكن نقله إلى حدوث آخر لهذه الفئة إلا كمقياس لقوة عمياء إذا حذف من قاعدة البيانات وأعيد إدخاله بمصاحبة حدوث فئة جديد.

إذا كان نوع السجل إجباري mandatory لفئة معينة فيجب أن يكون في سلسلة بعض حدوث لهذه الفئة دائماً وليس من الضروري أن يكون هذا الحدوث هو الحدوث المصاحب له في الأصل.

إذا كان نوع السجل اختياري optional بالنسبة إلى فئة معينة فيمكن فصله عن حدوث الفئة التي كان موجوداً فيه ويظل موجوداً في قاعدة البيانات ولا تعاد مصاحبته لحدوث آخر لهذه الفئة.

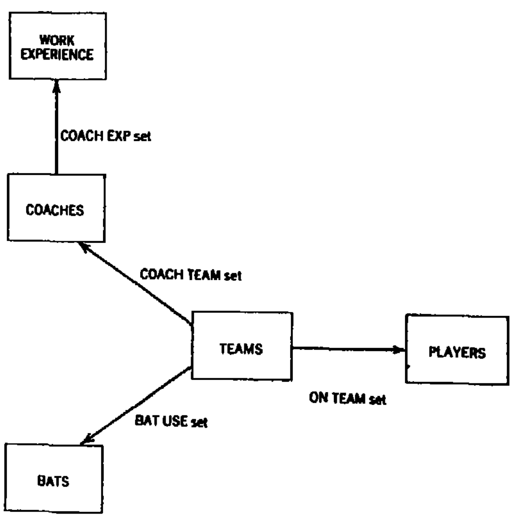
الحالية: في مناقشة IMS طورنا مفهوم المحافظ على الموقع في النظام أو مفهوم المشير الداخلي الذي يحفظ دائماً الإحساس بالموقع الحالي في قاعدة البيانات. وقد طور هذا المفهور تطويراً مرتفعاً في مواصفات CODASYL ويمكن حفظ عدد

كبير جداً من مثل مشيرات الموضع الحالي هذه كل منها يشير إلى تكوين جزئي من تكوينات الشبكة. تذكر أن النظام يحفظ فئة مختلفة من مشيرات الموضع لكل مستفيد لكل برنامج في نفس الوقت.

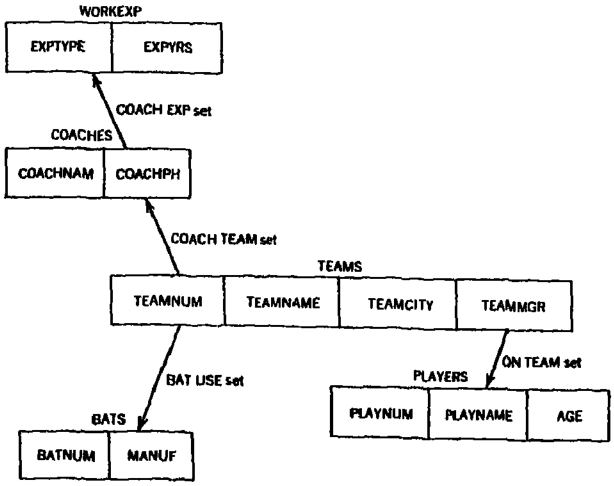
ومكافىء CODASYL لمشير الموضع الحالي الأساسي في IMS يسمى بحالية وحدة التشغيل current of run - unit. ويحدد بصفة دائمة آخر حدوث سجل (أي نوع سجل) يتم الوصول إليه في الشبكة. وتحدد حالية السجل current of record آخر حدوث لنوع سجل معين يتم لمسه. وهناك مثل هذا المؤشر لكل نوع سجل مختلف. وحالية الفئة current of set هي لآخر حدوث سجل يتم الوصول إليه في فئة معينة، مرة أخرى هناك مشير واحد من هذا النوع لكل فئة وكلها تكون محفوظة في نفس الوقت. حالية الحقل current of realm تكون لآخر حدوث سجل يتم الوصول إليه داخل فئة كبيرة من الشبكة تعرف بأنها حقل realm أو منطقة area.

عبارة FIND:

أول عبارة من عبارات CODASYL نفحصها هي عبارة FIND. نناقش هذه العبارة وكل عبارات لغة CODASYL مع شبكة متوازية مع مثال IMS الذي سبق استخدامه في الفصل السادس. شكل 7.14 عبارة عن تكوين شبكة يمثل نفس البيانات مثل التكوين الشجري Teams الموجود في شكل 6.4. ويبين



شكل 7.14
شبكة CODASYL

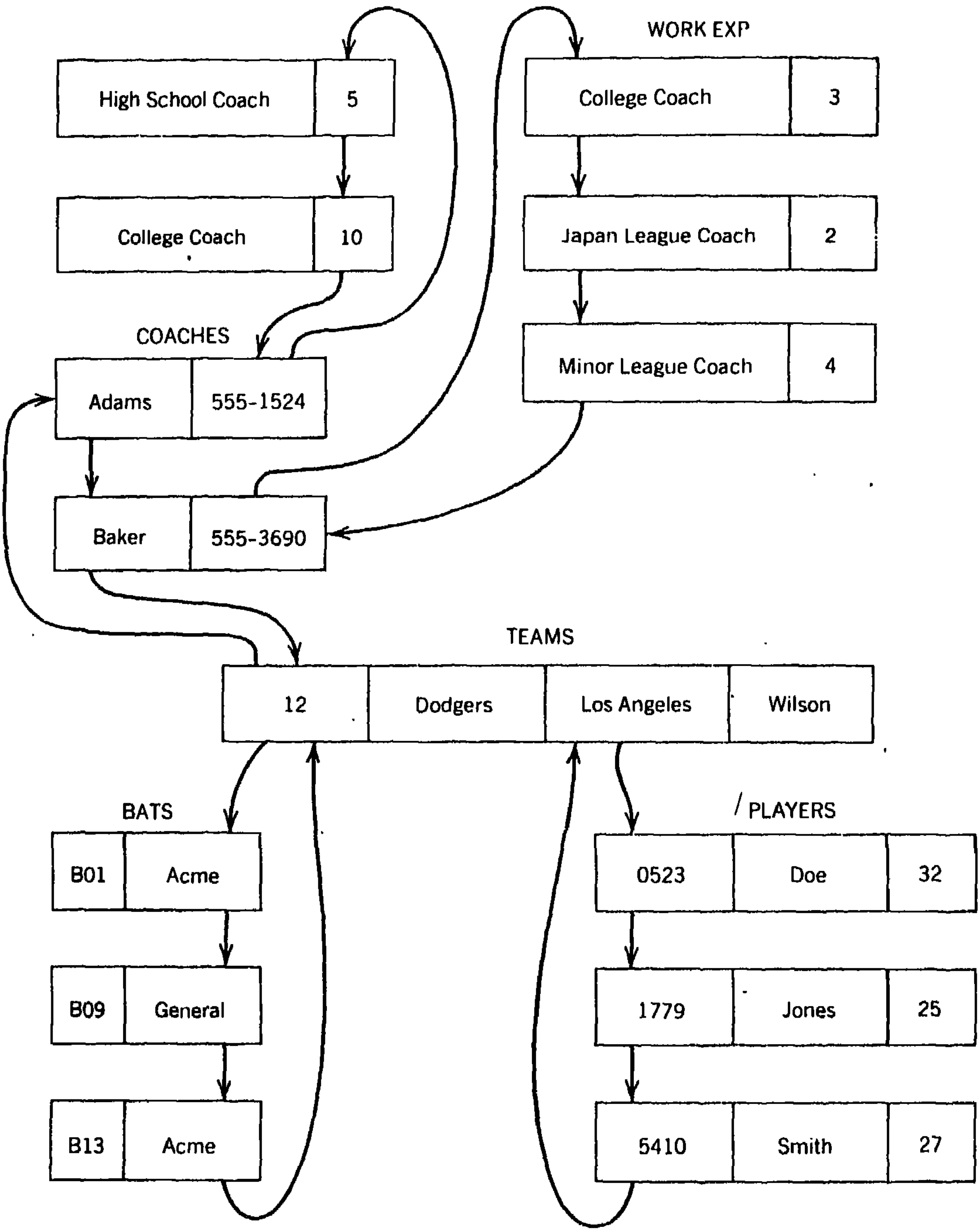


شكل 7.15
شبكة CODASYL الموجودة في شكل 7.14 مع بيان الأسماء

شكل 7.15 الحقول في سجلات تكوين الشبكة، هذا مثل ما فعل شكل 6.22 في حالة الهرمية. ويبين شكل 7.16 نفس البيانات الفعلية في الشبكة مثل ما فعل شكل 6.23 في الهرمية. والمشيرات في شكل 7.16 كلها مشيرات تالية.

كلمة تحذير قبل أن نبدأ: الصيغ المختلفة من مواصفات CODASYL وتنفيذات CODASYL المختلفة استخدمت تكوينات مختلفة للأوامر وهناك عائلتين رئيسيتين لمثل هذه التكوينات. سوف نحاول أن ندرس إحداها مع ذكر الأخرى بسرعة.

يمكن القول بصورة عامة أن عبارة FIND تجعل النظام يشير إلى حدوث سجل معين في قاعدة البيانات. وبصورة أكثر دقة فإن عبارة FIND توقع حدوث سجل معين وتضع مؤشر حالة وحدة التشغيل موجهاً له. (كما أنها تضع بصورة معتادة حالية السجل كذلك لنوع السجل المعني وحالية الفئة للفئة المعنية وحالية الحقل للحقل المعني). ولا تسترجع عبارة FIND السجل من المخزن فهذا يتطلب عبارة GET بعد عبارة FIND كما يظهر ذلك من المناقشة. وقبل أن تنفذ عبارة FIND يجب أن يضع البرنامج المتغيرات المناسبة في المخطط الجزئي المعرف في جزء البيانات من برنامج الكوبل وذلك لتحديد أي حدوث لنوع السجل مكتوب في عبارة FIND هو المراد توقيعه. وبهدف المقارنة تنفذ هذه العملية في IMS عن طريق قوائم بحث داخل الأمر نفسه.



شكل 7.16
حدوث جزئي للشبكة الموجودة في شكل 7.14

دعنا نبدأ بمثال بسيط لموقع سجل مباشر. إفرض أن سجلات TEAMS مخزنة بطريقة التكرار مباشرة والتي نسميها حالة calc لإيجاد سجل Dodgers في قاعدة البيانات فعليك أن تنقل MOVE القيمة 12 أولاً إلى حقل TEAMNUM في وصف سجل TEAMS في المخطط الجزئي ثم تصدر الأمر:

FIND ANY TEAMS.

وهناك صور مختلفة لهذا الأمر تشمل ما يلي:

FIND TEAMS RECORD

وكذلك ما يلي:

FIND ANY TEAMS USING TEAMNUM IN TEAMS

نتيجة لهذا الأمر فإن سجل Dodgers يصبح حالي لوحدة التشغيل وحالي السجل لنوع السجلات TEAMS وحالي الفئة للفئات ON TEAM و COACH TEAM و BAT USE وحالي الحقل لأي حقل موجود فيه.

فإذا كانت السجلات مخزنة في حالة CODASYL المباشرة وأنت تعرف معرف الموضع الفريد لحدوث سجل معين فعليك أن تصدر أمراً يشبه ما يلي:

FIND TEAMS; DB-KEY IS identifier.

تدعم مواصفات CODASYL كذلك مفهوم إيجاد حدوث السجل داخل حدوث فئة معينة. لعمل ذلك فإن أول شيء يتم عمله هو إعداد حدوث الفئة المعني. وأسهل طريقة لعمل ذلك هي تحديد حدوث السجل المالك للفئة المطلوبة. فإذا كان نوع سجل معين هو نوع السجل المالك لفئة معينة فإن كل حدوث لنوع هذا السجل المالك يعد حدوثاً للفئة.

وكمثال إفرض أنك تريد معلومات عن المضارب التي يستخدمها Dodgers. أول شيء تقوم بعمله هو أن تختار حدوث لفئة BAT USE يكون فيها Dodgers عبارة عن حدوث سجل مالك. يمكنك أن تفعل ذلك بعمل عبارة FIND بسيطة لسجل Dodgers والتي تعد ضمن أشياء أخرى حدوث (واحد وواحد فقط) لفئة BAT USE والتي يكون فيها Dodgers موجوداً على أنه حالي

فئة. بعمل ذلك، إذا أردت على سبيل المثال أول سجل BATS مع Dogers فعليك أن تصدر الأمر:

FIND FIRST BATS RECORD WITHIN BAT USE

حيث BATS هو اسم نوع السجل و BAT USE هو اسم الفئة. إذا أردت آخر المضارب المصاحبة لـ Dodgers فعليك أن تستبدل FIRST بكلمة LAST. أما كلمة NEXT فإنها تنقلك إلى التالي في السلسلة (قارن ذلك بأمر GET NEXT في IMS) وكلمة PRIOR تقيدك للسابق والرقم الصحيح الذي يمثل كثابت أو كمتغير صحيح يعطيك رقم الحدوث (first و second وthird . . . الخ) مع العد من أول حدوث سجل عضو في الفئة (والعودة للخلف إذا كان الرقم سالباً).

كما يمكنك أن تجد حدوث سجل كذلك داخل حدوث فئة مبنياً على قيمة أحد حقول حدوث هذا السجل.

على سبيل المثال إذا أردت أن تجد سجل مضارب Dodgers المسمى B09 فعليك أن تعد حدوث فئة BAT USE الموجود فيها Dodgers كسجل مالك كحالي للفئة وعند ذلك تصدر الأمر:

FIND BATS WITHIN BAT USE USING db-id

حيث db-id يعرف B09 على أنه مفتاح البحث. ماذا يكون الوضع إذا أردت أن تجد كل مضارب Dodgers التي تعدها Acme Company. إنك لا تستطيع استخدام صيغة الأمر التي سبق مناقشتها لأنه إذا كان هناك العديد من حدوثات سجل BATS لها نفس قيمة حقل البحث فإنك تجد في كل مرة يصدر هذا الأمر فيها أول حدوث له قيمة الحقل هذه. ويكون هناك حاجة لضرورة أخرى للأمر ومن الواضح أن الأمر يجب أن يوضع داخل دورة في برنامج لتشغيل كل حدوثات السجل بقيمة مفتاح البحث المحددة (كل مضارب لـ Dodgers والتي تعدها Acme Company). عليك أن تعد حدوث الفئة المناسب أولاً (مرة أخرى حدوث فئة BAT USE الموجود فيه Dodgers) لمضرب مستخدماً أحد تكوينات الأمر التي سبق مناقشتها. بعد ذلك عليك أن تجد بقية المضارب من

شركة Acme لـ Dodgers ولهذا فإنك تصدر ما يلي داخل دورة في البرنامج .

FIND DUPLICATE WITHIN BAT USE USING db-id

حيث db-id يعد مفتاح البحث.

ويجب أن توجد صيغة أخرى لعبارة FIND داخل فئة لتوقيع حدوث السجل المالك للفئة الحالية على النحو التالي:

FIND OWNER WITHIN BAT USE.

بالنسبة للنظم التي تدعم مفهوم الحقل فهناك أوامر شبيهة لإيجاد حدوثات السجلات داخل الحقل. ومشمولاً في هذه الأوامر إيجاد FIRST و LAST و NEXT و PRIOR أو رقم صحيح يعرف حدوث السجل لنوع سجل معين داخل الحقل.

أخيراً قد يكون من الضروري إعادة إعداد حدوث سجل كحالية لوحدة التشغيل. والاصطلاح إعادة إعداد يستخدم لأنه في هذه الحالة يكون قد سبق إيجاد السجل ويكون عند هذه اللحظة حالياً لسجل معين أو فئة معينة أو حقل معين. ولكن بسبب عبارات FIND المتتالية والتي تشمل سجلات أخرى لا يكون عند هذه اللحظة حالياً لوحدة التشغيل. مثال ذلك لإيجاد آخر مضرب مشار إليه في فئة BAT USE. فعليك أن تصدر الأمر:

FIND CURRENT BATS WITHIN BAT USE

أو الأمر:

FIND CURRENT OF BAT USE SET.

كملاحظة أخيرة تذكر كما كان الحال في IMS أن الشروط الاستثنائية يجب أن تختبر بعد إصدار أمر CODASYL. أمثلة هذه الشروط تشمل الوصول إلى نهاية الفئة أو إلى نهاية realm أوعدم وجود سجل (يتواءم مع معيار الاختيار المحدد) أو ان حالية وحدة التشغيل ليست نوع السجل الصحيح وما إلى ذلك.

عبارة GET:

تحضر عبارة GET حدوث السجل أو الحقول المختارة لحدوث السجل في داخل منطقة عمل المستفيد (user work area (UWA في الذاكرة الرئيسية. إذا ما أصدرت ببساطة الأمر التالي:

GET

فإن النظام يسترجع حدوث السجل الذي يكون حالياً لوحدة التشغيل. أما إذا أصدرت الأمر التالي لاسترجاع سجل TEAMS على سبيل المثال:

GET TEAMS

فلن يسترجع النظام حالية وحدة التشغيل فقط لكنه يختبر كذلك لمعرفة ما إذا كان في الواقع حدوثاً لنوع سجل TEAMS أم لا. بينما توجد إمكانيات وحدود مختلفة في النظم المختلفة فمن العدل عمل مقارنة واسعة بأن عبارة FIND في CODASYL التي يتبعها عبارة GET تحقق نفس الشيء الذي تحققه عبارة Get Unique أو عبارة Get Next (طبقاً للظروف) وذلك في IMS.

عبارات حفظ السجلات:

تخدم عبارات STORE و ERASE و MODIFY كأوامر حفظ للسجلات في قاعدة البيانات (بالرغم مما سوف تراه من أنه هناك آثاراً جانبية لهذه الأوامر تؤثر على عضوية الفئة). بعد هذا القسم سوف نناقش أوامر صيانة الفئات.

عبارة STORE:

تدخل عبارة STORE سجلات جديدة في قاعدة البيانات. يجب أن ينقل البرنامج البيانات المناسبة للسجل الجديد أولاً في UWA ثم يصدر على سبيل المثال الأمر التالي:

STORE TEAMS

لإدخال سجل TEAMS جديد. بالطبع يعالج النظام التفاصيل اللازمة للإدخال طبقاً لما إذا كانت السجلات من نوع السجل المعني مخزنة بطريقة تكرارية أو بفهرس أو بعنوان نسبي.

تذكر كذلك أن النظام يقوم تلقائياً بإضافة حدوث السجل إلى الفئات التي يسهم نوع السجل هذا فيها (فئات COACH و TEAM و BAT USE و ON TEAM للاستمرار في المثال).

إذا كان نوع تخزين هذا السجل في هذه الفئات موضحاً بأنه تلقائياً automatic في المخطط (إرجع إلى المناقشة السابقة الخاصة بنوع التخزين). كلمة تحذير: لعمل الاتصالات التلقائية بالفئات للحدوثات الصحيحة لهذه الفئات فيجب أن تحدد حدوثات الفئات هذه قبل إصدار أمر STORE. وهناك صيغاً مختلفة لكيفية عمل ذلك (يأتي ذلك تحت عنوان معروف بأنه اختيار الفئة set selection). إلا أنه يمكن عمل ذلك بوضع متغيرات في المخطط أو بإصدار عبارات FIND لتحديد مؤشرات الفئة الحالية.

عبارة ERASE:

تحذف عبارة ERASE سجلات من قاعدة البيانات في الصيغ المختلفة والتنفيذات المختلفة لمواصفات CODASYL. كان هناك عدد من صيغ لعبارة ERASE ومعها قيود مختلفة وسوف نناقش اثنين من هذه الصيغ.

إفرض أن إحدى الفرق خرجت من المجموعة. بالإضافة إلى السجل TEAMS وطبقاً للظروف فقد تريد أو لا تريد أن تحذف سلالات هذا السجل والمسماة بسجلات BATS و PLAYERS و COACHES و WORKEXP كما هو مبين في شكل 7.15. على سبيل المثال فإنك تجعل Dodgers حالية وحدة التشغيل ثم تصدر الأمر:

ERASE ALL TEAMS

بعد ذلك فإن Dodgers وكل حدوثات سلالة السجل أي كل شيء مبين في شكل 7.16 يحذف من قاعدة البيانات. أما إذا أصدرت من ناحية أخرى الأمر التالي:

ERASE Dodgers

بدون المؤشر «ALL» فإن قواعد نوع الحذف لنوع السجل في الفئة تؤخذ في الاعتبار. دعنا نقول على سبيل المثال أن BATS في فئة BAT USE ثابتة. هذا

يعني أن المضارب المحددة لـ Dodgers لا يمكن أن يستخدمها أي فريق آخر وكإحساس عام فسوف تحذف مع حذف Dodgers. إفرض أن اللاعبين players في فئة ON TEAM اختياريين. على ذلك يمكن أن يوجد اللاعبون مستقلون في قاعدة البيانات دون أي تحديد لفريق وعلى هذا فإنها تؤخذ خارج فئة ON TEAM وإلا فإنها تترك بمفردها. ثم ماذا عما إذا كان COACHES الموجودين في فئة COACH TEAM اجبارياً؟ عند ذلك يمكن نقل المدربين من فريق لآخر لكنهم يجب أن يظلوا في إحدى الفرق دائماً. وحيث أن عبارة ERASE Dodgers لا تقل أي الفرق سيعاد اتصاله بمدربي Dodgers فإن هذا يسبب مشكلة. وفي واقع الأمر فإن هذا يلغي سريان محتويات الأمر تاركاً قاعدة البيانات كما كانت من قبل.

عبارة MODIFY:

يمكن تحقيق تحديث السجلات المخزنة باستخدام عبارة MODIFY. مثال ذلك إذا أردت أن تغير بعض البيانات في سجل TEAMS فإنك تصدر أمر FIND ثم أمر GET لإحضار السجل في UWA ثم تقوم بتغيير حقول البيانات المناسبة ثم تصدر بعد ذلك الأمر التالي:

MODIFY TEAMS

عبارات حفظ الفئة:

تخدم العبارات CONNECT و DISCONNECT و RECONNECT كأوامر لحفظ الفئة كما يلي.

عبارة CONNECT

تصل عبارة CONNECT حدوث سجل موجود وهو حالي وحدة التشغيل في حدوث الفئة التي يكون حالية الفئة (لنوع الفئة المطلوب). مثال ذلك ماذا تفعل إذا كان اللاعب Stanton الموجود لكنه حالياً لاعب بدون عمل كلاعب، يلتحق بـ Dodgers؟ أولاً عليك أن تصدر أمر FIND لوضع مؤشر حالية الفئة للحدوث لفئة ON TEAM والتي عندها Dodgers كحدوث سجل مالك. بعد

ذلك عليك أن تصـدر أمر FIND لجعـل Stanton حاليـة وحدة التشغيـل وأخيراً فإنك تصدر الأمر التالي :

CONNECT PLAYERS TO ON TEAMS.

عبارة DISCONNECT:

إذا كان السجل حـالياً عبـارة عن حدوث سجـل عضو لحـدوث فئة معينـة فيمـكـن فصـله DISCONNECT مـن هـذه الـفئـة مـع اسـتـمـرار وجـوده في قـاعدة البيـانات سـواء كان مشمـولاً أم لا في أي فئة أخـرى. مثال ذلـك طرد Stanton من Dodgers يمكن أن تجعـل Stanton حاليـة وحـدة التشغيـل أولاً ثم تصدر الأمر التالي:

DISCONNECT PLAYERS FROM ON TEAMS.

نذكر قواعد نوع الإزالة. لجعل العبارة السابقة لها معنى فيجب أن يكون لها نوع إزالة محدداً اختيارياً لهذه الفئة.

عبارة RECONNECT:

تسمح عبارة RECONNECT بأن تنقل حـدوث سجل من حـدوث واحد للفئة إلى حدوث آخر لنفس الفئة. إفـرض على سبيـل المثال أن Stanton مـوجود حـالياً في Dodgers إلا أنـه سينقل إلى Giants. عليـك أن تعرف الحـدوث لفئـة ON TEAM أولاً الذي يوجـد فيه Giants كحـدوث سجل مـالك كحـدوث فئة هدف (معتمداً عـلى قواعـد اختيار الفئـة). بعد ذلـك عليك بـأن تجعل Stanton حالية وحدة التشغيل ثم تصدر الأمر التالي:

RECONNECT PLAYERS WITHIN ON TEAMS.

مـرة أخـرى خيـار نـوع الإزالـة المحـدد يجب أن يؤخـذ في الاعتبـار. ومن الـواضح أن عبـارة RECONNECT تكون في تعـارض مبـاشر مـع نـوع الإزالـة الثابت لنوع السجل هذا في هذه الفئة. ويجب أن تكون إجبارية أو اختيارية.

مراجع: References

CODASYL Journal of Development, January, 1978, CODASYL Data Description Language Committee, may be ordered from: Materiel Data Management Branch, Department of Supply and Services, 4th Floor, Core B1, Place du Portage, Phase III, II Laurier Street, Hull, Quebec, KIA 0S5, Canada.

Date, C. J., An Introduction to Database Systems, 3rd ed., vol. 1, Addisson-Wesley, Reading, MA, 1981.

King, J.M., Evaluating Data Base Management Systems, Van Nostrand Reinhold, New York, 1981.

Kroenke, D., Database Processing, 2nd ed., Science Research Associates, Chicago, IL, 1983.

Martin, J., Computer Data-Base Organization, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1977.

Olle, T.W., The Codasyl Approach to Data Base Management, Wiley, New York, 1978.

Ross, R. G., Systems, AMACOM, New York, 1978.

أسئلة وتمارين: Questions and Exercises

١ ـ صف الشبكة. صف جزءاً أو نشاطاً في إحدى المؤسسات يمكن أن يوصف باستخدام شبكة.

٢ ـ ما الفرق بين الشجرة والشبكة؟ وأيهما يعتبر فئة جزئية من الآخر؟

٣ ـ ما فئة CODASYL؟ ما نوع البيانات وعلامات البيانات التي تستطيع الفئة حملها؟ ما خيارات المشير في تخزين حدوثات الفئة؟

٤ ـ صف الصيغة الأساسية لشبكة CODASYL. ماذا تمثله العقد؟ كيف تسهم الفئات في التكوين؟ افتراضياً كيف تعالج المصاحبات المتعددة؟ وكيف تقدم قليلاً في تكرار البيانات؟.

٥ ـ أعد رسم حدوث جزء من الشبكة الموجود في شكل 7.9 ليعكس ما يلي (ملاحظة: اللاعبون مرتبون طبقاً للترتيب الأبجدي).

أ - يلحق Doe و Smith بـ Stanford.
ب - الولاية التي يقطن بها Doe هي New York.
جـ - يلعب Taylor و Turner لـ Giants.

٦ - ما سجل الاتصال؟ كيف تعالج علاقات من العديد للعديد في شبكات CODASYL؟

٧ - لماذا تكون سجلات الاتصال ضرورية فعلاً؟ ألا يمكن لعلاقات من العديد إلى العديد أن تعالج في شبكات CODASYL بدون هذه السجلات؟ وضح إجابتك.

٨ - ما المخططات وما المخططات الجزئية في نظم إدارة قواعد البيانات CODASYL؟ ما الفرق بينهما وما مميزات عمل هذا الفرق؟

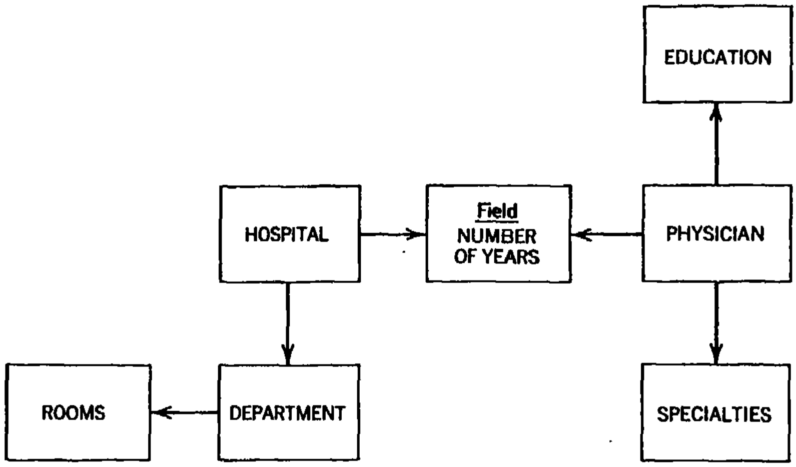
٩ - ما خيارات طرق الاتصال لتخزين شبكات CODASYL لإحدى النقاط في شبكة؟ ما التبادل trad - off بين هذه الخيارات؟

١٠ -كيف يمكنك أن تعدل شبكة Teams و Players الموجودة في شكل 7.13 لتؤدي ما يلي:
أ - تحفظ معلومات عن الحاصلين على تذاكر الموسم لكل فريق.
ب - تحفظ معلومات عن مهارات كل مدرب.
جـ - تحفظ بيانات مختلفة عن الكليات الملتحق بها اللاعبون (العديد من اللاعبين يمكن أن يكونوا ملتحقين بكلية معينة ولاعب معين يمكن أن يكون ملتحق بأكثر من كلية واحدة).

١١ - صف معنى شبكة CODASYL المبينة بالنسبة لبيئة العالم الواقعي التي تمثل الشبكة (ما طبيعة البيانات التي يمكن أن تخزن فيها وما نوع التطبيقات التي يمكن أن تستخدمها)؟

١٢ - ناقش مفاهيم نوع التخزين ونوع الإزالة في شبكات CODASYL ما هذه المفاهيم؟ وكيف ترتبط ببعضها؟ ولماذا تكون ضرورية ؟

١٣ - ما عبارات معالجة البيانات الرئيسية في مواصفات CODASYL؟ وهل تعطي هذه العبارات كل أنواع عمليات معالجة البيانات التي قد نحتاجها؟



١٤ ـبالإشارة إلى حدوث شبكة CODASYL الموجودة في شكـل 7.15 والمبين في شكل 7.16 حدد نتائج كل من تسلسل عبارات CODASYL التالية :

a. MOVE 12 to the TEAMNUM field in the TEAMS record
FIND ANY TEAMS
GET

b. MOVE 12 to the TEAMNUM field in the TEAMS record
FIND ANY TEAMS
FIND FIRST COACHES RECORD WITHIN COACH TEAM SET
GET

c. MOVE 12 to the TEAMNUM field in the TEAMS record
FIND ANY TEAMS
FIND COACHES WITHIN COACH TEAM USING Adams
GET

d. MOVE 12 to the TEAMNUM field in the TEAMS record
FIND ANY TEAMS
بداية الدورة
FIND NEXT PLAYERS RECORD WITHIN ON TEAM
GET
نهاية الدورة

١٥ ـ بالإشارة إلى حدوث شبكة COACHES الموجودة في شكـل 7.15 والمبينة في

شكل 7.16 اكتب استدعاءات CODASYL بدورات إذا كان ذلك ضرورياً لعمل ما يلي: (أنظر النهج المتبع في التمرين السابق).

أ ـ الحصول على سجل للمدرب Baker.

ب ـ الحصول على سجل أول خبرة للمدرب Baker.

جـ ـ الحصول على كل مدربي Dodgers.

هـ ـ إضافة لاعب جديد إسمه Turner إلى Dodgers.

و ـ حذف اللاعب Jones من قاعدة البيانات.

ر ـ تحديث عمر اللاعب Jones إلى 26 سنة.

ز ـ إزالة اللاعب Jones من Dodgers.

س ـ نقل اللاعب Jones من Dodgers إلى Giants.

١٦ ـما هي عبارات حفظ سجلات CODASYL؟ وهل تعطي هذه العبارات مدى كاملاً لأنشطة حفظ السجلات؟

١٧ ـناقش عبارات حفظ فئات CODASYL ماذا تفعل هذه العبارات؟ ولماذا هي ضرورية؟

الفصل الثامن

المنهج العلاقي لقاعدة البيانات

The Relational Approach to Database

مقدمة : **Introduction**

نشر دكتور ايدجركود Edgar F. Codd من شركة IBM في عام 1970 مقالاً بعنوان نموذج علاقي لبيانات خاصة ببنوك بيانات يشترك فيه الكثيرون. «A relational Model of Data for Large Shared Data Banks» لقد كانت هذه المقالة بداية لمجال قاعدة البيانات العلاقية. أثناء سنوات التأكد حصل المنهج العلاقي لقاعدة البيانات على كم هائل من الدعاية. ولم يصبح هناك نظم إدارة قواعد بيانات علاقية متاحة تجارياً إلا في السنوات القليلة الماضية. وقد كانت قاعدة البيانات العلاقية مغرية جداً في مفهومها إلا أنها كانت محيرة جداً حتى وقت قريب جداً عندما استخدمت في بيئة العالم الواقعي. وهناك العديد من الناس الذين يعتقدون أن هذا المنهج هو المنهج الذي يعد بالكثير جداً في المستقبل.

إن قصة نظم إدارة قواعد البيانات العلاقية من أكثر القصص الممتعة في مجال قاعدة البيانات الذي مازال يافعاً. كيف يمكن مقارنتها مع نظم إدارة قواعد البيانات الهرمية والشبكية بالنسبة للتشغيل والأداء والفلسفة العامة ليس ممتعاً فقط لكنه مفيد جداً للفهم الحقيقي لبعض المفاهيم الأساسية لقاعدة البيانات.

نبدأ هذا الفصل بقاعدة أساسية سهلة الفهم في المفاهيم والعمليات العلاقية. بعد ذلك نعد بعض مقارنات بين المنهج العلاقي والمنهج الإبحاري. بعد ذلك نصف لغة قاعدة بيانات علاقية واسعة الانتشار والاستخدام. وأخيراً نفحص نظام آخر مقام وهو نظام إدارة قاعدة بيانات علاقية بمنهج جديد لتداخل المستفيد.

مفاهيم قاعدة البيانات العلاقية:

Relational Database Concepts

مراجعة لمقدمة قاعدة البيانات:

لقد بدأنا مناقشة قاعدة البيانات في الفصل الرابع بملفات خطية بسيطة منفصلة في شكل 4.1 وملف مبرمج في شكل 4.2. وللإقناع فقد أعيد هذين الشكلين في شكلي 8.1 و 8.2. دعنا نراجع الموقف الخاص بالملفات في هذين الشكلين. الملفان الموجودان في شكل 8.1 عبارة عن ملفين خطيين بسيطين. وحقل SALESPERSON NUMBER هو الحقل الرئيسي الفريد لملف البائعين. أما حقل CUSTOMER NUMBER فهو الحقل الرئيسي الفريد لملف العملاء. هناك سجل واحد لكل بائع في ملف البائعين وسجل واحد لكل عميل في ملف العملاء. وهذان الملفان لهما أعلى خاصية مرغوبة ليكونا بسيطين افتراضياً. وبالإضافة إلى ذلك لا يوجد تكرار بيانات بين حقولهما غير الرئيسية لأي بائع يرد اسمه ورقمه (وافتراضياً عنوانه ورقم هاتفه. . . إلخ) مرة واحدة بالضبط في ملف البائعين. ويمكن قول نفس الشيء على بيانات العملاء في ملف العملاء.

المشكلة مع هذين الملفين هي ما يبدو من عدم المقدرة على تكامل البيانات في أي طريقة موجزة بالقوة الغاشمة لبرمجة التطبيقات. لإيجاد اسم name البائع المسؤول عن أحد العملاء فيجب على مبرمج التطبيق أن يجد سجل هذا العميل في ملف العملاء ثم يستخلص رقم البائع منه ويستخدم قطعة البيانات هذه كمدخلات لملف البائعين لإيجاد السجل الخاص بهذا البائع والذي يوجد به اسمه.

الاختيار الآخر وهو الملف المدمج في شكل 8.2 والذي يحل مشكلة تكامل البيانات. يمكن الإجابة على الاستفسار السابق باسترجاع سجل واحد - ولكن مع إدخال تكرار كثير وغير مقبول للبيانات.

ونؤكد أن نظام إدارة قاعدة البيانات الحقيقي يجب أن يوفر تكويناً للبيانات يسمح بتخزين البيانات في صورة غير متكررة كما يجعلها تظهر كذلك لمبرمج التطبيق بأن البيانات متكاملة تكاملاً رفيعاً. لقد بينا في الفصلين السادس والسابع

SALESPERSON NUMBER	. . .	SALESPERSON NAME	. . .
137		Baker	
186		Adams	
204		Dickens	
361		Carlyle	
⋮		⋮	

(a)

CUSTOMER NUMBER	. . .	SALESPERSON NUMBER	. . .
0121		137	
0839		186	
0933		137	
1047		137	
1525		361	
1700		361	
1826		137	
2198		204	
2267		186	
⋮		⋮	

(b)

شكل 8.1

(أ) ملف البائعين (ب) ملف العملاء

SALESPERSON NUMBER	CUSTOMER NUMBER	SALESPERSON NAME	. . .
137	1021	Baker	
137	0933	Baker	
137	1047	Baker	
137	1826	Baker	
186	0839	Adams	
186	2267	Adams	
204	2198	Dickens	
361	1525	Carlyle	
361	1700	Carlyle	
⋮	⋮	⋮	

شكل 8.2

ملف مدمج

كيفية تحقيق هذا الهدف من خلال استخدام تكوينين غير خطيين للبيانات: الهرميات والشبكات.

إذا أوقفت تفكيرك فيها فإن مقدرة نظم إدارة قواعد البيانات الهرمية والشبكية على حفظ عدم تكرار للبيانات وفي نفس الوقت على تقديم جو متكامل كان ناتجاً مباشراً لتكوينات البيانات المستخدمة. وكما رأينا في الفصلين السادس والسابع تمسك الهرميات والشبكات ببيانات غير متكررة. فيمكن تخزين بيانات أحد فرق البيسبول مرة واحدة فقط ثم تصاحب بعد ذلك كل من اللاعبين الموجودين على سبيل المثال في الفريق عن طريق أن سجلاتهم متصلة ببعضها البعض. هذا الاتصال المتداخل للسجلات يوفر إحساساً كذلك بتكامل البيانات والتي يمكن أن يتمتع المبرمج بهذه الميزة وذلك بالإبحار خلال التكوينات بالأوامر المناسبة في برامج تطبيقاته.

حل آخر لمشكلة قاعدة البيانات:

ما العلاقة؟ بينما تمثل حلول الهرمية والشبكة للمشكلة قاعدة لاستقطاب الكثير من السوق العالمي حالياً إلا أن الكثير من الناس قد أرهقوا من البساطة النسبية للملفات الخطية البنسيطة الموجودة في بيئة قواعد بياناتهم. وقد فكروا فيما إذا لم يكن من الأفضل أن يكونوا قادرين على معاملة البيانات كما لو كانت مخزنة في ملفات خطية بسيطة مع استمرار تقدم النظام لنوع معين من إمكانية تكامل البيانات. وهذه الرغبة هي القوة الدافعة بالضبط لقواعد البيانات العلاقية.

لنبدأ الموضوع دعنا نعتبر تكوين البيانات المستخدم في قواعد البيانات العلاقية. وهذا سهل. كما سبق القول تظهر البيانات بأنها مخزنة فيها نشير إليه بأنه ملفات خطية بسيطة. وسواء كان التخزين الواقعي مرتباً بهذه الطريقة أم لا أو أنه يجعلها تظهر بهذا الشكل فهذا لا يهم. باتباع اصطلاحات مجال الرياضيات بأن قواعد البيانات العلاقية موجودة فسوف نبدأ بتسمية هذه الملفات الخطية البسيطة بعلاقات relations أو بجداول tables. نفترض أنه يوجد صفين أو سجلين أو إثنين taples من علاقة متطابقين. وتقنياً فإن الأعمدة أو الحقول أو الخواص attributes الخاصة بعلاقة يمكن أن ترتب بأي ترتيب دون التأثير على معنى

البيانات. والعلاقة دائماً لها حقل أو مجموعة حقول تخدم كمفتاح فردي لها. ويجب أن يكون هذا صحيحاً حيث أننا نعرف أنه لا يوجد صفين يمكن أن يكونا متطابقين وبالتالي كأسوأ حالة فإن كل حقول العلاقة يمكن أن تخدم كمفتاح إذا كان ذلك ضرورياً (إلا أن هذا الحل نادر الحدوث في الحياة العملية). بهذا المفهوم يكون من المقنع التفكير في المفاتيح بالنسبة الى كونها حقول فردية. وسوف نفعل ذلك حالياً إذا كانت العلاقة لها أكثر من حقل فردي واحد كمفتاح فيقال عليها جميعها أنها مفاتيح مرشحة Candidate Keys. والمفاتيح الذي يختار منها مفتاح ليكون مفتاحاً للعلاقة يسمى بالمفتاح الابتدائي primary key. وإذا خدمت إحدى الخواص في مجموعة من العلاقات التي تمثل قاعدة بيانات علاقية كحقل فردي لعلاقة واحدة وظهرت كحقل في علاقة أخرى فتسمى بمفتاح أجنبي fore-ign key في هذه العلاقة الأخيرة. ومفهوم المفتاح الأجنبي مفهوم مهم وسوف نوضحه باختصار.

على هذا فقاعدة البيانات العلاقية هي ببساطة عبارة عن تجميع لعلاقات. بالنسبة إلى الطرق التي نفكر بها في البيانات فإن العلاقات تكون مستقلة على عكس التكوينات المتصلة اتصالاً محكماً بالمشيرات الموجودة في منهجيات قواعد البيانات الهرمية والشبكية. بافتراض أن العلاقات مثل الموجودة في شكل 8.1 (سوف نبدأ الإشارة إلى هذين الملفين على أنهما علاقتين الآن)، مصممة جيداً فلا يكون هناك تكرار في الحقول غير الرئيسية (الحقول الأخرى غير المفاتيح). وسوف يترك السؤال إلا أننا سوف نضع التعليقات المناسبة عن ذلك كلما كان هناك حاجة إلى ذلك في هذا الفصل.

بالمناسبة وكما هو الحال في أي منهج من مناهج قواعد البيانات فهناك موضوع طرق الاتصال المبني على التكوين العلاقي. حيث أن العلاقة يتم التفكير فيها أساساً على أنها ملف خطي بسيط فأي طريقة من طرق الاتصال التي سبق مناقشتها في هذا الكتاب تسري عليها صفوف العلاقة يمكن أن تخزن بطريقة التكرار أو بفهرس أو بترتيب تتابعي بسيط (بدون فهرس). سوف نرى أن طبيعة العمليات العلاقية سوف تجعل من استخدام طريقة الاتصال التتابعية طريقة غير مقبولة عملياً. كما سيبدو واضحاً كذلك أن الكفاءة في نظم إدارة قواعد البيانات العلاقية عادة ما تعتمد على الاتصال المباشر لعدد من الخواص الخاصة بالعلاقة.

وعلى هذا فنظم إدارة قواعد البيانات العلاقية تسمح عادة بانتاج الفهارس عبر العديد من أو كل الخواص الخاصة بالعلاقة.

استخلاص بيانات من العلاقة: بوجود قاعدة بيانات علاقية تحتوي على علاقة واحدة أو أكثر فإننا نبدأ في التفكير في كيفية استخلاص بيانات منها. بدون أي معدات خاصة لاستخلاص بيانات من العلاقة تتحول المشكلة ببساطة الى كتابة برنامج بإحدى لغات البرمجة لاسترجاع بيانات من ملف خطى بسيط. إذا كان هذا هو كل ما نستطيع عمله فلا يكون هناك في الواقع مجال للكلام عن قاعدة بيانات علاقية. إلا أن النقطة الحاسمة هي أن السمة الداخلية لنظام إدارة قاعدة بيانات علاقية هي المقدرة على قبول أوامر استرجاع بيانات مرتفعة المستوى تهدف ملفات خطية بسيطة أو علاقات وتقوم بتشغيلها حتى إلى مدى السجلات المتواءمة الموجودة في ملفات مختلفة كما في حالة إيجاد اسم name البائع المسؤول عن حساب أحد العملاء. دعنا نرى ما يمكن أن يكون شكل هذا القول بالنسبة إلى علاقة فردية أولاً ثم عبر علاقات متعددة.

حيث أن العلاقة هي ترتيب مستطيلي لمجموعة من القيم فقد يبدو ملموساً أنه يمكن المعالجة أفقياً أو رأسياً أو أفقياً ورأسياً. أخذ شريحة أفقيه من علاقة يشمل استرجاع صفاً واحد أو أكثر من صفوف العلاقة. وهذا لا يزيد عن كونه لغة خيالية لاسترجاع سجل واحد أو أكثر من ملف. أما أخذ شريحة رأسية من علاقة فإنه يعني استرجاع صفة واحدة (أو خاصية واحدة) أو أكثر من كل صفوف العلاقة أو عمود واحد أو أكثر من كل صفوف العلاقة. ومبرمج المعالجة الأفقية والرأسية يمكنه استرجاع خاصية أو أكثر لأحد الصفوف أو لأكثر من صف واحد من العلاقة وأقل شيء في هذه الحالة هو الخاصية الفردية لصف فردي أو الحقل الفردي لسجل فردي. وهذا احساس استرجاع طيب كما نريد.

باستخدام اصطلاح من مصطلحات تكوين قواعد البيانات يسمى بجبر العلاقات relational algebre وشكل أمر حر حالياً دعنا ننظر إلى أمرين يسميان اختر Select وأسقط Project وهما قادران على المعالجة الأفقية والرأسية التي سبق ذكرها. بالنسبة لقاعدة البيانات فإننا نستخدم العلاقتين الموجودتين في شكل 8.3 وهما نفس العلاقتين الموجودتين في شكل 8.1 فيما عدى أننا وضعنا الآن بعض

SALESPERSON NUMBER	SALESPERSON NAME	COMMISSION PERCENTAGE	YEAR OF HIRE
137	Baker	10	1975
186	Adams	15	1971
204	Dickens	10	1963
361	Carlyle	20	1963

(a)

CUSTOMER NUMBER	SALESPERSON NUMBER	CITY
0121	137	New York
0839	186	Hartford
0933	137	Boston
1047	137	Boston
1525	361	Newark
1700	361	Washington
1826	137	New York
2198	204	New York
2267	186	New York

(b)

شكل 8.3
(أ) علاقة البائعين
(ب) علاقة العملاء

الحقول غير الرئيسية وحذفنا النقاط التي تحدد وجود صفوف وخواص غير معروفة. لنبدأ، افرض أننا نريد إيجاد سجل البائع رقم 204. بطريقة مباشرة جداً يمكن أن يكون الأمر كما يلي: اختر Select صفوفاً من علاقة البائعين Salesperson. يكون فيها SALESPERSON NUMBER = 204.

SALESPERSON NUMBER	SALESPERSON NAME	COMMISSION PERCENTAGE	YEAR OF HIRE
204	Dickens	10	1963

لاحظ أن نتيجة عملية الاختيار Select هي في حد ذاتها علاقة وتشمل في هذه الحالة صفاً واحداً فقط. نتيجة العملية العلاقية دائماً تكون علاقة سواء كانت تحتوي على عدة صفوف بها العديد من الخواص أو على صف واحد به خاصية واحدة فقط (أي على قيمة واحدة فقط).

ربما نريد كل السجلات التي لها قيمة مشتركة في أحد حقولها (غير الرئيسية) المحددة. عند ذلك نكتب ما يلي:

اختر Select صفوفاً من علاقة البائعين Salesperson يكون فيها COM-MISSION PERCENTAGE = 10. نتيجة هذه العملية هي ما يلي:

SALESPERSON NUMBER	SALESPERSON NAME	COMMISSION PERCENTAGE	YEAR OF HIRE
137	Baker	10	1975
204	Dickens	10	1963

بالطبع قد نريد استرجاع محتويات الملف وربما يكون ذلك بأمر يشبه ما يلي:

اختر Select كل الصفوف الموجودة في علاقة البائعين SALESPERSON.

لاسترجاع ما سبق أن أشرنا إليه بشريحة رأسية فإننا نستخدم مؤثر الإسقاط Project. لاسترجاع رقم واسم كل بائع في الملف فإننا نصدر الأمر التالي:

Project the SALESPERSON NUMBER and SALESPERSON NAME over the Salesperson relation.

والنتيجة هي علاقة ضيقة طويلة على النحو التالي:

SALESPERSON NUMBER	SALESPERSON NAME
137	Baker
186	Adams
204	Dickens
361	Carlyle

إذا أسقطنا حقلاً غير رئيسياً فيجب أخذ قرار عند مستوى معين بما إذا كنا نريد ازدواجاً أم لا (بالرغم من عدم استطاعتنا تقنياً عمل ذلك حيث أن النتيجة في حد ذاتها عبارة عن علاقة).

Project the YEAE OF HIRE over the Salesperson relation

ينتج عنه ما يلي:

YEAR OF HIRE
1975
1971
1963
1963

أو

YEAR OF HIRE
1975
1971
1963

يعتمد ذلك على التنفيذ الخاص المستخدم.

ما زال خليط مؤثرات الاختيار Select والإسقاط Project أكثر قوة. إفرض أننا طبقنا الأمرين على التوالي مع العلاقة التي تنتج من عملية واحدة تستخدم كعلاقة تعمل عليها العملية التالية.

دعنا نقول أننا نريد أرقام وأسماء البائعين الذين يعملون بعمولة 10%. نصدر الأمرين التاليين لهذا الغرض:

Select tuples from the Salesperson relation in which COMMISSION PERCENTAGE = 10. Project the SALESPERSON NUMBER and SALESPERSON NAME over that result.

وينتج عن ذلك ما يلي:

SALESPERSON NUMBER	SALESPERSON NAME
137	Baker
204	Dickens

والخليط التالي يوضح المقدرة على استرجاع قيمة فردية. إفرض أننا نريد أن نعرف سنة تعيين البائع رقم 204. حيث أننا نعرف أن SALESPERSON NUMBER هو حقل فردي فإننا نعرف أننا نتعامل مع صف واحد فقط. وحيث أننا نبحث عن خاصية واحدة من هذا الصف فإننا نعرف أننا نريد قيمة فردية:

Select tuples from the Salesperson relation in which SALESPERSON NUMBER = 204. Project the YEAR OF HIRE over that result.

والنتيجة هي القيمة الفردية التالية:

YEAR OF HIRE
1963

استخلاص بيانات من علاقات متعددة: إننا ندخل في مناقشة عن تكامل البيانات في منهج قاعدة البيانات العلاقية من خلال محتوى استخلاص البيانات من علاقات متعددة. أنظر إلى العلاقات الموجودة في شكل 8.3 وفكر مرة أخرى في المثال السابق الخاص بتكامل البيانات: إيجاد اسم البائع المسؤول عن حساب أحد العملاء. الحل المقترح لهذه المشكلة حتى الآن كان عن طريق إيجاد سجل حساب العميل في ملف العملاء وقراءة رقم البائع من هذا السجل ثم استخدام هذا الرقم في إيجاد سجل البائع الموجود في ملف العاملين.

ألن يكن أظرف إذا كان نفس النظام القادر على استرجاع بيانات مبنية على أنواع أوامر Select و Project والذي سبق إن تعرضا له يكون قادراً كذلك على أداء هذا النوع من البحث في علاقة متعددة؟ في الواقع فإنه باستخدام مؤثر يسمى توصيل Join مع Select و Project يمكنا حل مشكلة تكامل بيانات العلاقة المتعددة.

بالاستمرار مع بنفس أسلوب الأوامر المسترخاة دعنا نصدر أوامر لإيجاد اسم البائع المسؤول عن العميل رقم 1525 على النحو التالي:

Join the Salesperson relation and the Customer relation, using the SALESPERSON NUMBER of each as the join field. Select tuples from that result in which CUSTOMER NUMBER = 1525. Project the SALESPERSON NAME over that result.

دعنا نفحص أول جملة والتي تمثل أمر وصل. تذكر كيفية استخلاص قيمة خاصية SALESPERSON NUMBER الموجودة في علاقة واحدة واستخدامها في البحث عن السجل المطلوب في علاقة أخرى. لقد كنا قادرين على عمل ذلك بدقة لأن هذا الحقل كان موجوداً في كل من العلاقتين. (حتى إذا كانت الخاصيتين لهما اسمين مختلفين مثل SALESPERSON NUMBER في إحدى العلاقات و EMPLOYEE NUMBER في العلاقة الأخرى فما زال في إمكاننا تنفيذ نفس العملية لأنهما يمثلان نفس نوع المعلومات.

ويمكننا أن نقول من الناحية التقنية أنهما لهما نفس نطاق القيم (domain of values). عملية الوصل تأخذ مميزات الحقل المشترك SALESPERSON NUMBER (والذي يسمى بالنسبة لهذا الأمر بحقل الوصل Join field). وينظر إلى الصفوف في العلاقتين التي لها نفس القيمة في حقل الوصل. وعندما يجد زوجاً من السجلات المطلوبة فإنه يأخذ كل حقول السجلين وينتج سجلاً جديداً واحداً منها في العلاقة الناتجة. وهذه العملية مرهقة في الواقع حيث تتم مقارنة صف أحد العلاقات بصفوف العلاقة الأخرى والبحث عن مواءمة في حقول الوصل ونتيجة أمر الوصل وهو أول أمر من الثلاثة أوامر المتتالية التي تنفذها هي ما يلي:

SALES-PERSON NUMBER	SALES-PERSON NAME	COMMISSION PERCENTAGE	YEAR OF HIRE	CUSTOMER NUMBER	SALES-PERSON NUMBER	CITY
137	Baker	10	1975	0121	137	New York
137	Baker	10	1975	0933	137	Boston
137	Baker	10	1975	1047	137	Boston
137	Baker	10	1975	1826	137	New York
186	Adams	15	1971	0839	186	Hartford
186	Adams	15	1971	2267	186	New York
204	Dickens	10	1963	2198	204	New York
361	Carlyle	20	1963	1525	361	Newark
361	Carlyle	20	1963	1700	361	Washington

لاحظ أن أول وسادس خاصية متطابقتين. وتمثلان خاصية SALESPERSON NUMBER من علاقتي البائعين والعملاء على التوالي. تذكر أن الصفين لا يدمجان معاً لتكوين صفاً واحداً في العلاقة الناتجة إلا إذا كانت قيمة هذه الحقلين (حقلي الوصل) متطابقة في أول مكان. وهذا النوع من الوصل يسمى الوصل المتساوي equi - join. وكما يبدو أنه معقولاً إذا حذفنا أحد العمودين المتطابقين في العملية فإننا نسمي النتيجة الوصل الطبيعي natural join.

للاستمرار بتسلسل الأوامر لإيجاد اسم البائع المسؤول عن العميل 1525 فإننا نصدر أمر Select أولاً الذي سبق لنا كتابته مستخدمين العلاقة الناتجة من الوصل. وينتج عن ذلك ما يلي:

SALES-PERSON NUMBER	SALES-PERSON NAME	COMMISSION PERCENTAGE	YEAR OF HIRE	CUSTOMER NUMBER	SALES-PERSON NUMBER	CITY
361	Carlyle	20	1963	1525	361	Newark

وأخيراً فإننا نصدر الأمر الثالث project لنحصل على ما يلي.

SALES PERSON NAME
Carlyle

لاحظ أنه كان يمكننا أن نجعل هذه العملية سلسلة. لقد كنا نتعامل مع عميل واحد فقط وكان هناك صف واحد فقط لكل عميل في علاقة العملاء له حقل رئيسي واحد هو CUSTOMER NUMBER. وعلى هذا كان علينا أن نبحث في الواقع عن صف واحد في علاقة العملاء لحل المشكلة. سوف نتكلم عن استخدام مثل هذه المعلومات لزيادة الكفاءة الكلية للعملية فيما بعد في هذا الفصل وذلك داخل إطار نظام لغة الاستفسار المرتبة Structured Query Language System.

هناك وجه آخر لمفهوم الوصل وهو الوصل الخارجي outer join (على عكس الوصل الذي سبق ذكره والمسمى بالوصل الداخلي inner join). في الوصل الخارجي تظهر كل صفوف العلاقة في العلاقة الأخرى. فإذا لم يتواءم حقل وصل لصف معين مع أي صف آخر في العلاقة الأخرى فإن هذا الصف يظهر في العلاقة الناتجة وله قيم صفرية في مكان القيم من صف المواءمة في العلاقة إذا كان هناك واحد.

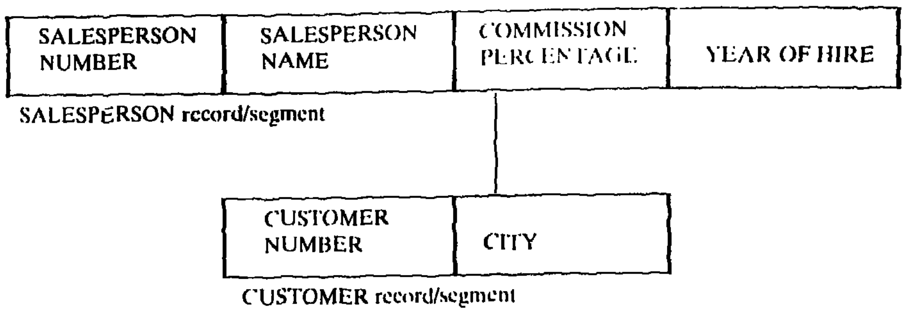
المقارنة بين المنهج العلاقي والمناهج الأخرى:

نقاط المقارنة: أحد طرق مقارنة المنهج العلاقي بالمناهج الأخرى هي من خلال مفهوم الوصل. بينما تكلمنا حتى الآن عن الوصل فقط في محتوى العلاقات إلا أنه يمكننا أن نناقش ذلك مع العودة إلى الأمور المرتبطة به جداً من تكامل البيانات في محتوى النظم الهرمية والشبكية كذلك. هناك طريقة أخرى للمقارنة وهي تصميم قاعدة بيانات وتعديلها أثناء استخدامها.

هناك طريقة ثالثة لمقارنة المناهج العلاقية وغير العلاقية وذلك عن طريق

خواص الأداء. وسوف توضح هذه المناقشة أسباب أن قواعد البيانات العلاقية أصبحت حالياً فقط مهمة تجارياً. وهناك طريقة رابعة للمقارنة تشمل توضيحات التكامل عند وقت عمليات الإدخال والحذف.

الوصل هو الوصل هو الوصل. يبين شكل 8.4 نفس نموذج بيانات البائعين والعملاء موجوداً في صورة هرمية أو شبكية. المقارنة بين العلاقات الموجودة في شكل 8.3 وتكوين شكل 8.4 مهمة جداً وتقود إلى قلب مفهوم قاعدة البيانات.



شكل 8.4

هرمية أو شبكة البائعين والعملاء

بغض النظر عن تكوين نظام إدارة قاعدة البيانات المستخدم في تطبيق معين فيظل هناك نفس المفهوم الأساسي لتكامل البيانات. يجب أن تكون السجلات التي تصف كينونات مختلفة في بيئة التطبيق قادرة بطريقة معينة على أن تصاحب بعضها البعض. وفي الأساس هذا ليس مفهوماً عميقاً في علم الكمبيوتر إلا أنه طريقة عمل العالم الذي نعيش فيه. فالبائع مسئول عن العديد من العملاء. والعميل لديه رقم بائع محدد له ليرسل إليه طلباته. والعالم يدور حول مثل هذه العلاقات. وعندما نحاول على ذلك عمل نموذج للعالم في بيئة موجهة للبيانات وتشغيل بيانات تلقائياً فإن هذه العلاقات تترجم إلى مفهوم تكامل البيانات.

عد إلى شكل 8.3 وفكر في كيف يمكن أن تظل هذه العلاقات التي تقود إلى تكامل بيانات في بيئة علاقية (أو لهذا الأمر ملف خطي بسيط). بالعلاقات المستقلة فإن تسجيل المعلومات التي تعد صفوفاً للعلاقة يكون لها مصاحبة ذات

معنى يجب أن يوجد لها صف (صفوف) في علاقة أخرى وتكون هذه الصفوف واضحة. من الواضح أن خاصية SALESPERSON NUMBER تنتمي إلى علاقة البائعين وفي الواقع فهي تظهر في هذه العلاقة. بالإضافة إلى ذلك بينما يكون هناك شيء نأخذه على أنه مضمون منذ الوهلة الأولى فإن حقيقة أن SALESPERSON NUMBER يظهر كذلك في علاقة العملاء Customer تكون حاسمة في الإحساس ببيئة التطبيق. أي أنه كيف نعرف أي العملاء يكون البائع مسئولاً عنهم وأي بائع يلجأ إليه العميل لتسليم الأمر الخاص به. لقد وضعت خاصية SALESPERSON NUMBER في الواقع في علاقة العملاء عندما صممت هذه التكوينات ولم يكن هذا موضوعاً جزافاً وسوف يناقش ذلك تحت عنوان تصميم قاعدة البيانات في الفصل العاشر من الكتاب.

والآن ركز انتباهك على الهرمية أو الشبكة المناظرة والموجودة في شكل 8.4. هناك كذلك بائعين مصاحبين لعملاء ولكن بطريقة مختلفة فهم مرتبطون ضمنياً بواسطة تصميم التكوين الهرمي ومرتبطون بوضوح بواسطة المشيرات التابعة والتوأم أو التالية والتي أعدت عندما تم تحميل البيانات الفعلية طبقاً للتكوين الذي سبق تحديده. لاحظ أن Customer Record في شكل 8.4 لا يوجد فيه حقل SALESPERSON NUMBER. ليس هناك حاجة لهذا الحقل عندما يتم تحميل البيانات طبقاً للتكوين المعد مسبقاً حين أن المشيرات الطبيعية physical pointers سوف تعد وهي التي تربط بعد ذلك العملاء بالبائعين المناظرين.

إلا أن هذا يقود إلى سؤال هام آخر. هل مفهوم الوصل موجود في قواعد البيانات الهرمية والشبكية؟ الإجابة كما يجب أن تكون قادراً على افتراضها عند هذه النقطة هي نعم (على الأقل بصورة غير منظمة). والآن فيما يلي نقطة أساسية في المقارنة بين قواعد البيانات العلاقية وغير العلاقية:

> في قواعد البيانات الهرمية والشبكية تنفذ الوصلات ضمنياً في الوقت الذي تصمم فيه الهرميات والشبكات وتصمم واضحة مع المشيرات عندما يتم تحميل البيانات طبقاً لهذه التكوينات. وقوة تكامل البيانات تكون أساساً دالة في تكوين البيانات.

في قواعد البيانات العلاقية تنفذ الوصلات على الهواء on the fly عند وقت التنفيذ بواسطة نظام إدارة قاعدة البيانات طبقاً للأوامر المعطاة للنظام عند ذلك الوقت. تكوين البيانات أي العلاقات يجب أن يكون لها حقول وصل موجودة في الأماكن الصحيحة. إلا أن القوة الأساسية لتكامل البيانات تكون في وقت التنفيذ وتكون وظيفة من وظائف نظام إدارة قاعدة البيانات.

تكامل البيانات هو مفهوم أساسي والوصل هو وسيلة تحقيق ذلك بطريقة أو بأخرى.

المرونة في التصميم والتعديل: كما سنرى في الفصل العاشر هناك طريقتان واضحتان وكذلك عدة طرق أقل وضوحاً لتصميم قاعدة البيانات. وكل طريقة لها مؤيديها الذين يشعرون بأن الطريقة التي يفضلونها هي الأسهل. وهذا في الواقع جزء من مفهوم أكبر عادة ما يثار عن أي مناهج قاعدة البيانات هو الأكثر واقعية بالنسبة إلى الطريقة التي يمثل بها مواقف تطبيقات العالم الواقعي. ومن المعروف أن خبراء قواعد البيانات يجادلون بالساعات في هذا السؤال دون الوصول إلى اتفاق لأن الموضوع يعتمد على الآراء الشخصية. فكل شخص يعتبر الأمر طبيعياً natural. وكلمة طبيعي ليست كلمة تقنية ولن نناقش هذا الموضوع.

وبالرغم من ذلك سوف نلاحظ في تمرير ذلك أنه هناك العديد من الأفراد الذين يشعرون بأن العلاقات أسهل في تصميمها وأنها أبسط داخلياً في إدارتها عن الهرميات أو الشبكات. أحد الأشياء التي تحدث عند تصميم قاعدة بيانات علاقية هي التوزيع المناسب للحقول الواصلة في العلاقات المبني على الطرق التي تعمل على أساسها بيئة التطبيق. وهذا يؤكد أن أي استفسار صحيح من ناحية دلالة الألفاظ (أي يكون له معنى بالنسبة للتطبيق) ويتطلب عملية وصل فإنه يجد حقول الوصل في الأماكن الصحيحة التي يحتاج أن يجدها فيها. إلا أن عمليات الوصل وبصفة خاصة وصلات القوى العمياء التي تتطلب فحصاً للملف كله أو للملفين كلهما فيمكن أن تكون بطيئة جداً للملفات الكبيرة.

عند تصميم قاعدة بيانات هرمية أو شبكية فيجب أن تؤخذ القرارات الخاصة بتكامل المصاحبات التي تصمم بالفعل داخل التكوين. وفي البيئات

الهرمية والشبكية يمكن أن يكون أداء الاتصال الموجه للوصل والمبني على العلاقات المصممة داخل التكوين (والتي تنفذ عادة بالمشيرات) سريعاً جداً بالرغم من أن الاتصال المبني على العلاقات التي لم تصمم داخل التكوين يمكن أن يكون بطيئاً جداً.

ويمكن في العادة عمل تعديلات مستقبلية على قاعدة البيانات وذلك طبقاً لطبيعتها ويتم ذلك بسهولة في قواعد البيانات العلاقية أيضاً. ونحن لا نتكلم هنا عن إضافة سجلات إضافية أو حدوثات لصفوف في قاعدة البيانات ولكننا نتكلم عن تغييرات أو توسعات في عدد أنواع types السجلات. مثال ذلك يمكن إضافة علاقة داخلية جديدة إلى قاعدة بيانات علاقية دون التأثير على صيغ الاستفسار الموجودة أو على العلاقات الموجودة. وقد يتطلب إضافة أنواع جديدة من السجلات إلى قاعدة البيانات الهرمية أو الشبكية أو توضع هذه الأنواع بالضبط داخل التكوينات الموجودة نفسها مع إعادة تحميل البيانات في قاعدة البيانات وإعادة تحديد أوضاع المشيرات وقد يؤثر ذلك على الاستفسارات في برامج التطبيقات الموجودة.

ولكنه من المهم كذلك ملاحظة أنه إذا كانت طبيعة التغيير تشمل تغييراً جذرياً في التطبيق مثل أن رقم البائع لا يصبح فريداً أو أن أكثر من بائع واحد يكون مسؤولاً عن أحد العملاء فعند ذلك قد تحتاج قاعدة البيانات إلى إعادة تصميمها بغض النظر عن المنهج المتبع مع قاعدة البيانات.

خواص الأداء: وأخيراً يأتي الوقت للإجابة على السؤال الخاص بسبب أن نظم إدارة قواعد البيانات العلاقية لم تصبح متاحة تجارياً إلا حديثاً. تذكر توضيح الوصل السابق في هذا الفصل والمثال والذي أوصلنا فيه علاقتين في شكل 8.3. ففي عينة العلاقات نجد أن علاقة البائع لها أربعة صفوف فقط وعلاقة العميل لها تسعة صفوف فقط. وبفكرة مقارنة كل صف لأحد هذه العلاقات مع كل صف للعلاقة الأخرى لتحقيق الوصل تبدو معقولة بحيث أنها تجعلك لا تفكر فيها مرة أخرى. لكن ماذا يحدث لو حاولنا وصل علاقة بها مليون صف بعلاقة بها 3 مليون صف. ما الوقت اللازم لذلك حتى إذا استخدم كمبيوتر سريع جداً؟

إن ذلك قد يأخذ وقتاً أكثر من الوقت الذي يتوقع الشخص أن ينتظر استجابة عبر النهاية الطرفية.

ومن ناحية أخرى ففي بيئة تطبيق قاعدة بيانات علاقية تقليدية تتعامل استفسارات عديدة لقاعدة البيانات مع علاقة واحدة فقط ولا تتطلب أي عملية وصل على الإطلاق. وما زال السؤال قائماً: ماذا عن الاستفسارات التي تتطلب ذلك؟ تعتمد قواعد البيانات الحديثة والمعقدة على وحدات نظم مكونات خاصة لزيادة الأداء أو على أساليب نظم برامج ماهرة في الإسراع في عملية الوصل عن طريق تقليل مدى الوصل بالنسبة الى عدد الصفوف المشمولة مثلاً. كما أنها تأخذ ميزة الزيادة العامة في سرعة مشغلات الكمبيوتر التي تتطور باستمرار عبر السنوات. سوف نناقش وحدات نظم المكونات في الفصل 14 من هذا الكتاب وأساليب نظم البرامج فيما بعد في هذا الفصل تحت عنوان لغة الاستفسار المرتبة.

وما الموقف المقارن مع قواعد البيانات الهرمية والشبكية؟ تدور النقطة الرئيسية هنا حول حقيقة أن تكامل البيانات يتحقق في وقت تحميل البيانات وحفظها بواسطة كل المشيرات الطبيعية التي تقوم بتوصيل السجلات. ويمكن أن تظهر الهرميات والشبكات على أنها تكوينات قواعد البيانات في المستقبل لكنك لا تستطيع أن تدفع المشيرات الطبيعية لزيادة سرعتها. وهذا هو الجانب الآخر للعملة: السبب على الأقل بالنسبة للأداء هو أن نظم إدارة قواعد البيانات المبنية على الهرميات والشبكات كانت ناجحة تجارياً منذ بداية عصر قواعد البيانات.

توضيح التكامل بالنسبة للحذف والإضافة: دعنا نقول أن شركة المثال تحصل على حساب عميل وتحدده لبائع معين. فإذا كان IMS هو نظام إدارة قاعدة بياناتها وكانت تكوينات البيانات كما هي ممثلة في شكل 8.4 فعند ذلك نستطيع أن نضيف قطاع CUSTOMER جديد فقط كقطاع لاحق بقطاع SALESPERSON المحدد والذي له معنى معين بالنسبة للمخطط العام للأشياء. ونستطيع في بيئة الشبكات أن نقحم مواصفات الآلية automatic على الموقف كمؤشر إدخال فئة. أما في علاقات شكل 8.3 فلا يوجد شيء داخلي في تكوين البيانات لاتمام صف خاص ببائع معين ليوجد في علاقة البائع Salesperson وذلك لأن هناك شخص معين يدخل سجل العميل في علاقة العميل Customer

والتي يكون مسجلاً فيه هذا البائع. وهذا توضيح التكامل الذي يجب أن يعالج في النظام أو برنامج التطبيق. والإصطلاح الرسمي لذلك هو التكامل الإشاري referential integrity. وبالمثل يمكن حذف سجل أحد البائعين تاركاً قيمة تشير إليه في سجل العميل مع عدم وجود شيء يشار إليه.

وبهذه المناسبة لاحظ أنه تحت ظروف تطبيقات معينة فإن مقدرة إضافة صف إلى علاقة بغض النظر عن اعتبارات التكامل الإشاري يمكن أن يمثل إحدى الميزات. مثال ذلك قد نريد أن نضيف عميل جديد إلى ملف العملاء وذلك بدون تحديد البائع المحدد والذي لأي سبب من الأسباب لا يكون ميسراً حالياً في ملف البائعين. وبالرغم من ذلك فهناك الكثيرون الذين يشعرون أنه لكي يغطي نظام إدارة قاعدة بيانات علاقي مدى كاملاً لنظم تطبيقات عملية فإن التكامل الإشاري يكون متطلباً أساسياً لنظم إدارة قواعد البيانات العلاقية في المستقبل.

لغة الاستفسار المرتبة: Structured Query Language (SQL)

مقدمة:

كمثال للغة قاعدة بيانات علاقية سوف نفحص لغة الاستفسار المرتبة SQL لشركة IBM. وهذه هي لغة علاقية مستخلصة من مشروع أبحاث سابق لشركة IBM في قواعد البيانات العلاقية يسمى النظام R «System R». وهو مبرمج حالياً في نظامين لإدارة قواعد البيانات لشركة IBM وهما: لغة استفسار مرتبة/ نظام بيانات SQL/DS والذي يعمل تحت نظامي التشغيل DOS و VM الحاليين وقاعدة البيانات 2 (DB2) والذي يعمل تحت عائلة MVS لنظم التشغيل الكبيرة. كما أنه هناك شركات أخرى أعدت نظم إدارة قواعد بيانات مبنية على SQL أيضاً.

وبالحفاظ على طابع هذا الكتاب فإننا لن نهتم بالاختلافات الموجودة بين تنفيذات لغة الاستفسار المرتبة SQL (فهي بسيطة نسبياً على أية حال) ولكننا نهتم بالقواعد الأساسية والتكوينات العامة الموجودة فيها، وسوف نستخدم اصطلاح SQL للإشارة الى اللغة والنظام الشامل. ونلاحظ كذلك أثناء المرور أن كل من SQL/DS و DB2 عبارة عن نظام إدارة قاعدة بيانات كامل مع وجود احتياطيات

لعمل نسخ احتياطية ولعمليات الاستعادة بعد الفقدان واختبارات الأمان وما إلى ذلك ولكن هذه المواضيع لن تناقش في هذا الفصل. وكما سبق أن فعلنا في الفصول السابقة عن مفاهيم قواعد البيانات الهرمية والشبكية فسوف نركز على أوجه اللغة وتركيبها المرتب هنا ونؤجل مناقشة المفاهيم الأخرى إلى فصل لاحق. (لاحظ أن الموضوع التالي عن تكوين بيانات SQL وامثليتها مطبق على تنفيذات IBM للغة SQL وليس سمة داخلية للغات SQL بصفة عامة).

تكوين بيانات SQL:

هناك نقطتان أريد الإشارة لهما بالنسبة لتكوين بيانات وهما: إن البيانات تبدو أنها مخزنة كملفات خطية بسيطة أو كعلاقات. وباصطلاحات SQL تسمى جداول.

ويتم الاتصال بالجداول إما تتابعياً أو عن طريق فهارس. ويمكن أن يشير الفهرس إلى عمود واحد أو أكثر (وباصطلاحات SQL يستخدم عمود بدلاً من حقل) من الجدول. ويمكن أن يكون للجدول العديد من الفهارس المعدة له. ويمكن أن يشير الفهرس إلى عمود تكون قيمته فريدة أو غير فريدة. وعندما تتغير البيانات الموجودة في الجدول فتجدد SQL البيانات المناظرة في أي فهرس والتي تتأثر بهذا التغيير ويحدث ذلك بصورة تلقائية.

وينفذ مفهوم قواعد البيانات المنطقية والواقعية في SQL ويسمى الجدول الواقعي بجدول الأساس base table. كما يسمى الجدول المنطقي بالمنظر (رؤية) view وهو عبارة عن جزء من جدول أساسي أو جزء من جداول أساس. ويمكن أن يحتوي المنظر على جزء من الأعمدة أو جزء من الصفوف الموجودة في الجدول. كما يحدث في تنفيذات مفهوم قاعدة البيانات المنطقية فإن انتاج المنظر بلغة SQL لا يعني إزدواجاً واقعياً للبيانات الموجودة في جدول بوضعها في جدول آخر. وبدلاً من ذلك تحفظ المعلومات التي تصف طبيعة هذا الجزء في أحد كتالوجات النظام العديدة Catalogs، فإذا أشار أحد الاستفسارات إلى أحد المناظر (يمكن إصدار الاستفسارات لجداول الأساس أو للمناظر) فتقوم المعلومات الخاصة بالمنظر برسم من المنظر إلى الجدول الواقعي حيث توجد البيانات المطلوبة في التخزين.

أمثليه SQL:

يسبق الاستجابة لاستفسار المستفيد أن تحاول SQL رسم طريقة أكثر كفاءة للإجابة على هذا الاستفسار وبصورة غير رسمية فمضمون ما يحدث هو ما يلي:

SQL موجهة ناحية عمل تعظيم لمميزات فهارسها. وهذا يجعلها بالطبع معتمدة على الشخص الذي يصمم جداول SQL لتحتوي على فئة مناسبة من الفهارس المبنية على الأنواع المتوقعة من الاستفسارات. دعنا نستطرد من عند هذه النقطة. تذكر أولاً وقبل كل شيء أن SQL سوف تجيب على استفسارات بغض النظر عما إذا وجدت أو لم توجد فهارس يمكنها أن تسرع من الأشياء. ثانياً تذكر أن أعداد خليط لعدد كبير من الفهارس قد يؤثر على الأداء. فإذا كانت قاعدة بيانات SQL قابلة التقلب أي أنها لها كم معنوي من أنشطة الإدخال والحذف والتجديد فعند ذلك تميل الإمكانية التلقائية الموجودة في SQL التي تجدد الفهارس الى أن تبطيء النظام الكلي - كلما زادت الفهارس كلما حدث بطء للنظام.

ويقوم القائم بعمل الأمثلية optimizer بالاختبار لمعرفة إذا كان الاستفسار يشمل أعمدة لها فهارس مبنية عليها أم لا. فإذا كان هناك ذلك فإنه يلاحظ العوامل الأخرى الخاصة بقيم الأعمدة. وللعديد من عمليات اختيار النوع يكون من الواضح أخذ مميزات هذه الفهارس والذهاب مباشرة إلى السجلات المطلوبة فقط. فإذا كان هناك وصل في الاستفسار فيحاول النظام أن يرسم الطريقة التي يمكن أن تحصل على أقصى مميزات لأي فهارس مناسبة. مثال ذلك إذا طلب الاستفسار عملية وصل وكانت أعمدة الوصل (حقول الوصل) في كل جدول في ترتيب معين أولها فهارس مبنية عليها فعند ذلك لا يكون ضرورياً مقارنة كل سجل من سجلات أحد الجداول مع كل سجل من جدول آخر. فيمكن للنظام أن يبدأ ببساطة عند قمة كل جدول (أو الفهرس إذا لم يكن الجدول مرتباً بواسطة عمود الوصل) ويتحرك لأسفل متتبعاً أثر القيمة الحالية في كل من الجدولين كما لو كان يؤدي عملية دمج. ويشمل مثال آخر الحالة التي يكون فيها الوصل متطلباً سجلاً فردياً فقط من أحد الجداول والذي يجب أن يتواءم مع سجل أو عدة سجلات موجودة في جدول آخر.

ومن الواضح أنه إذا كان هذا السجل من الجدول الأول يمكن أن يوجد بصورة مباشرة بأحد الفهارس فلا يكون النظام في حاجة إلى المرور خلال كل سجلات هذا الجدول كما هو الحال في وصله القوة الغاشمة. ومثالنا الأصلي الخاص بإيجاد اسم البائع المسؤول عن حساب عميل معين هو أحد هذه الحالات.

لاحظ أنه في المرور بهذا المثال يتأكد النظام كذلك من أن الأعمدة المحددة موجودة في واقع الأمر في الجداول المحددة. أي أن المستفيد يكون له السلطة لرؤية هذه البيانات وأن الأعمدة العددية لا تقارن مع أعمدة حرفية عددية وما إلى ذلك. وتخزن كل هذه المعلومات في مجموعة من كتالوجات النظام.

لغة SQL

سوف نستكشف في هذا القسم العديد من أوامر لغة SQL في مضمون مثال البيسبول الذي سبق ذكره. تمثل الستة جداول الموجودة في شكل 8.5 معظم تكوين المثال الأصلي. إحصائيات اللاعبين وبيانات المواقع غير موجودة ولكن من ناحية أخرى يوضح الكثير من حدوثات السجلات في بقية التكوين. والتغييرين أجريا بغرض تعليمي ولا يشملان شيئاً عن إمكانية النظام. قارن شكل 8.5 مع شكل 6.15 لاحظ كيف تناظر جداول TEAMS و COACHES و WORKEXP و BATS و PLAYERS المقاطع الموجودة في تكوين IMS بصورة مباشرة. جدولي TEAMS و PLAYERS لهما أعمدة رئيسية فردية (حقول رئيسية فردية) وهي TEAMNUM و PLAYNUM على التوالي. وحيث أنه من الممكن وجود مدربين اثنين مختلفين مع آخر اسم في فريقين مختلفين فيتم الحصول على الفردية في جدول COACHES عن طريق خليط من أعمدة TEAMNUM و COACHES وبعد ذلك يظهر أن الفردية في جدول WORKEXP تحتاج إلى خليط من أعمدة TEAMNUM و COACHNUM و EXPTYPE حيث أنه بالإضافة إلى التعليق السابق يمكن أن يكون للمدرب العديد من قطع الخبرة. أرقام المضارب تكون فردية داخل الفريق فقط وعلى هذا فإن أعمدة TEAMNUM و BATNUM تكون لازمة لفردية جدول BATS. وجدول AFFILIATION يمثل في قاعدة البيانات العلاقية علاقة من العديد إلى العديد مبيناً تاريخ اللاعبين، أي الفرق قاموا باللعب لها أثناء لعبهم، وتاريخ

الفرق، أي اللاعبين كانوا بهذه الفرق. لاحظ كيف تمثل أعمدة YEARS و BATAVG الموجودة في جدول AFFILIATION نفس بيانات التقاطع -in

TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
12	Dodgers	Los Angeles	Wilson
15	Giants	San Francisco	Johnson
20	Yankees	New York	Simpson
24	Tigers	Detroit	Corbin

TEAMS Table

TEAMNUM	COACHNAM	COACHPH
12	Adams	555-1524
12	Baker	555-3690
15	Taylor	555-4820
20	Jackson	555-2444
24	Victor	555-9327
24	Dooley	555-7321

COACHES Table

TEAMNUM	COACHNAM	EXPTYPE	EXPYRS
12	Adams	High School Coach	5
12	Adams	College Coach	10
12	Baker	College Coach	3
12	Baker	Japan League Coach	2
12	Baker	Minor League Coach	4
15	Taylor	College Coach	15
24	Victor	Minor League Coach	12

WORKEXP Table

TEAMNUM	BATNUM	MANUF
12	B01	Acme
12	B09	General
12	B13	Acme
15	B01	Acme
20	B03	United
20	B04	Modern
24	B18	General
24	B21	United

BATS Table

شكل 8.5 قاعدة بيانات علاقية لمثال البيسبول

tersection data التي تظهر في مقاطع IMS المعتمدة المنطقية. أنظر إلى العـلاقات المـوجودة في شكـل 8.5 واقنع نفسـك بأن البيـانـات ليست مكـررة. فكـل فـريق موصوف مرة واحدة فقط وكذلك كل لاعب وما إلى ذلك.

PLAYNUM	PLAYNAME	AGE
358	Stevens	21
523	Doe	32
1131	Johnson	28
1779	Jones	25
2007	Dobbs	27
4280	Cohen	25
4319	Ross	24
5410	Smith	27
6564	Linton	24
8093	Smith	21
8366	Gomez	33

PLAYERS Table

PLAYNUM	TEAMNUM	YEARS	BATAVG
358	15	5	0.300
358	20	3	0.320
523	12	10	0.257
1131	20	1	0.283
1779	12	1	0.223
1779	15	7	0.246
1779	24	2	0.240
2007	24	3	0.290
4280	15	1	0.195
4280	20	3	0.227
4319	15	4	0.298
5410	12	6	0.307
6564	20	12	0.310
6564	24	3	0.280
8093	12	5	0.250
8093	20	2	0.240
8093	24	8	0.265
8366	20	7	0.283

AFFILIATION Table

شكل 8.5 (تابع)

يمكن إصدار أوامر SQL كأوامر استفسار خاصة adhoc أو كأوامر اتصال محتواة في برامج مكتوبة بلغة الكوبل أو الفورتران أو لغة PL/I أو لغة المجمع. ونفترض في هذا القسم أن SQL تستخدم في حالة استفسار خاص. والتكوين في الحالة المحتواة هو نفسه من الناحية الوظيفية مثل حالة الاستفسار مع بعض إضافات يمكن أن تتوقعها. وسوف نذكر مثال أو مثالين لهذه الاختلافات مع تقدمنا في توضيحات الأوامر. كما يجب أن تتحقق كذلك من أنه عندما يستخدم أحد أوامر SQL في الحالة المحتواة ويعيد علاقة متعددة كنتيجة له فعلى البرنامج أن يقوم بتشغيل هذه النتيجة صفاً واحداً أو سجلاً واحداً في نفس الوقت كما هو مطلوب.

استرجاعات جدول فردي باستخدام SQL

دعنا نبدأ ببعض أوامر اختيار وأنواع إسقاط بسيطة. وفي الواقع فإن كل أوامر أنواع الاسترجاع في SQL بما في ذلك الأوامر التي سبق الكلام عنها مثل الاختيار select والإسقاط project والوصل join سوف تنفذ بواسطة أمر SELECT. وللتوضيح فسوف نستخدم الترحيل مع وضع الأجزاء المختلفة من الأوامر في أسطر مختلفة بالرغم من أن أوامر SQL يمكن أن تكتب كسلسلة مستمرة عبر نهاية طرفية بها شاشة. ولاسترجاع محتوى جدول TEAMS عليك أن تصدر الأمر:

```
SELECT *
FROM TEAMS
```

وترجمة ذلك الحرفية تقول «إعرض كل (*) الأعمدة الموجودة في جدول TEAMS». والصورة التقليدية حيث أنه لم تتحدد أي قيود هي أن كل السجلات تكون مشمولة. وتشبه النتيجة جدول TEAMS الموجود في شكل 8.5 بالضبط (وهي مناسبة حيث أن نتيجة الاستفسار العلاقي تكون علاقة في حد ذاتها).

TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
12	Dodgers	Los Angeles	Wilson
15	Giants	San Francisco	Johnson
20	Yankees	New York	Simpson
24	Tigers	Detroit	Corbin

ولاختيار سجل فريد عن طريق تعريف قيمة عمود فردية مثل «أوجد سجل الفريق رقم 12». عليك أن تصدر الأمر:

```
SELECT *
  FROM TEAMS
  WHERE TEAMNUM = 12
```

ويستخدم جزء WHERE وذلك لوضع حد للشريحة الأفقية من الجدول متسبباً في استرجاع جزء من السجلات. وفي هذه الحالة فإن العمود المحددة قيمته يكون عموداً فريداً وبالتالي لا يسترجع إلا سجل واحد فقط.

TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
12	Dodgers	Los Angeles	Wilson

وعلى سبيل المثال ففيما يلي عملية إسقاط بسيطة «اسرد رقم الفريق واسم الفريق لكل الفرق. والتي تشبه ما يلي:

```
SELECT TEAMNUM, TEAMNAME
  FROM TEAMS
```

أسماء الأعمدة التي تأتي بعد SELECT مباشرة هي الأعمدة المراد استرجاعها:

TEAMNUM	TEAMNAME
12	Dodgers
15	Giants
20	Yankees
24	Tigers

وعلى سبيل المثال لدمج عملية اختيار وعملية إسقاط بسيطة أوجد رقم اللاعب واسمه لكل الاعبين الذين يزيد عمرهم عن 30 عاماً فإنك تصدر الأمر:

```
SELECT PLAYNUM, PLAYNAME
  FROM PLAYERS
  WHERE AGE>30
```

لاحظ أن جزء WHERE مرن جداً ويسمح بالمدى الكامل لمؤثرات المقارنة. وإذا ما أدمجت هذه العبارة في برنامج فيمكننا استبدال الثابت 30 بتغير توضع قيمته قبل ذلك في البرنامج والنتيجة في هذه الحالة تصبح:

PLAYNUM	PLAYNAME
523	Doe
8366	Gomez

ويمكن دمج الشروط المتعددة في عبارة SELECT فإذا أردت قائمة بالمدربين

الذين عملوا مدربين في كليات لأكثر من خمسة سنوات فإنك تكتب ما يلي:

```
SELECT TEAMNUM, COACHNAM
FROM WORKEXP
WHERE EXPTYPE = 'College Coach'
AND EXPYRS\5
```

والنتيجة تصبح كما يلي:

TEAMNUM	COACHNAM
12	Adams
15	Taylor

والإسقاط على عمود غير فردي يمكن أن ينتج عنه محتويات مزدوجة في النتيجة. مثال ذلك إذا أردت قائمة بالفرق طبقاً لرقم الفريق والتي تستخدم مضارب تضعها شركة Acme Company فعليك أن تكتب ما يلي:

```
SELECT TEAMNUM
FROM BATS
WHERE MANUF ='Acme'
```

وتكون النتيجة كما يلي:

TEAMNUM
12
12
15

وفي مثل هذه الحالة فأنت لست مهتماً في الازدواجات وعليك أن تضيف كلمة «distinct»:

```
SELECT  DISTINCT TEAMNUM
FROM  BATS
WHERE   MANUF = 'Acme'
```

لتحصل على ما يلي:

TEAMNUM
12
15

وفيما يلي مثال يوضح أن السجلات المختارة يمكن أن تقتصر على مدى قيم لعمود ويمكن أن ترتب بواسطة قيم أي عمود. للحصول على سجلات لاعبين تقع أعمارهم بين 25 و 29 سنة وترتيب هذه السجلات طبقاً للعمر عليك أن تكتب ما يلي:

```
SELECT *
FROM PLAYERS
WHERE AGE BETWEEN 25 AND 29
ORDER BY AGE
```

والنتيجة تكون كما يلي:

PLAYNUM	PLAYNAME	AGE
1779	Jones	25
4280	Cohen	25
2007	Dobbs	27
5410	Smith	27
1131	Johnson	28

ويمكن اختيار السجلات طبقاً لقيمة إحدى أعمدتها الموجودة في فئة محددة. فإذا أردت قائمة بالمضارب التي تم تصنيعها بواسطة شركة General Company أو شركة United Company والفرق التي تستخدمها فعليك أن تصدر ما يلي:

```
SELECT *
FROM BATS
WHERE MANUF IN ('General', 'United')
```

وتحصل على النتيجة التالية:

TEAMNUM	BATNUM	MANUF
12	B09	General
20	B03	United
24	B18	General
24	B21	United

ويمكن استرجاع السجلات طبقاً لمشاركتها في جزء فقط من قيم أحد الأعمدة. وعلى سبيل المثال لتحقيق الاستفسار «أوجد سجلات كل اللاعبين الذين يبدأ اسم العائلة لهم بالحرف S» فعليك أن تصدر ما يلي:

```
SELECT *
FROM PLAYERS
WHERE PLAYNAME LIKE 'S%'
```

وتكون النتيجة كما يلي:

PLAYNUM	PLAYNAME	AGE
358	Stevens	21
5410	Smith	27
8093	Smith	21

وهي معتمدة على إشارة النسبة المئوية التي تعوض عن أي شيء بعد الحرف .S

وهناك عدد من الدوال المبنية داخلياً تعطي SQL امكانيات إضافية للحصول على عدد السجلات الموجودة في جدول. على سبيل المثال لإيجاد عدد الفرق فيمكنك أن تصدر ما يلي:

```
SELECT COUNT *
FROM TEAMS
```

والنتيجة تظهر كما يلي:

COUNT(EXPRESSION 1)
4

وهناك صورة أخرى لهذا الأمر وهي عد عدد القيم المميزة في عمود معين من أعمدة الجدول. مثال ذلك إذا أردت إيجاد عدد المنتجين المختلفين للمضارب الذين يوردون للفرق فيمكنك أن تصدر ما يلي:

```
SELECT COUNT (DISTINCT MANUF)
FROM BATS
```

والتي ينتج عنها ما يلي:

COUNT(DISTINCT MANUF)
4

ويمكن توجيه العد باستخدام جزء WHERE أيضاً. وللإجابة على

الاستفسار أوجد عدد اللاعبين الذين يزيد عمرهم عن 30 سنة عليك بأن تصدر ما يلي:

```
SELECT COUNT(*)
    FROM PLAYERS
    WHERE AGE>30
```

والتي ينتج عنها ما يلي:

COUNT(EXPRESSION 1)
2

وهناك معالم خاصة أخرى لنفس الأمر العام مثل COUNT والتي تشمل إيجاد مجموع العديد من الأرقام وإيجاد متوسطها وأقصى وأقل قيمة لها. مثال ذلك لإيجاد إجمالي عدد سنوات خبرة المدرب Adams في Dodgers (الفريق رقم 12) فعليك أن تصدر ما يلي:

```
SELECT SUM(EXPYRS)
FROM WORKEXP
WHERE TEAMNUM = 12
AND COACHNAM = 'Adams'
```

ويستجيب النظام كما يلي:

SUM(EXPYRS)
15

وقد تريد أن تنفذ إحدى هذه العمليات على أجزاء عديدة من أجزاء السجلات الموجودة في الجدول مستقلة عن بعضها. مثال ذلك أنك قد تريد أن تعرف إجمالي عدد سنوات الخبرة السابقة لكل من المدربين. لإيجاد ذلك عليك أن تستخدم سمة «التجميع بواسطة» وتصدر ما يلي:

```
SELECT TEAMNUM, COACHMAN, SUM(EXPYRS)
FROM WORKEXP
GROUP BY TEAMNUM, COAGHNAM
```

وتكون النتيجة كما يلي:

TEAMNUM	COACHNAM	SUM(EXPYRS)
12	Adams	15
12	Baker	9
15	Taylor	15
24	Victor	12

استرجاعات SQL للجداول المتعددة (الوصلات)

دعنا نقدم أمر الوصل ثم نناقشه بعد ذلك. أمر الوصل بين جدولي COACHES و TEAMS يمكن أن يكتب على النحو التالي:

```
SELECT TEAMNUM, COACHNAM, SUM(EXPYRS)
  FROM WORKEXP
  GROUP BY TEAMNUM, COACHNAM
```

وينتج عنه ما يلي:

TEAMNUM	COACHNAM	COACHPH	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
12	Adams	555-1524	12	Dodgers	Los Angeles	Wilson
12	Baker	555-3690	12	Dodgers	Los Angeles	Wilson
15	Taylor	555-4820	15	Giants	San Francisco	Johnson
20	Jackson	555-2444	20	Yankees	New York	Simpson
24	Victor	555-9327	24	Tigers	Detroit	Corbin
24	Dooley	555-7321	24	Tigers	Detroit	Corbin

لاحظ النقاط التالية عن الأمر: يحدد التعبير From COACHES, TEAMS أنه هناك جدولان معنيان في هذا الأمر والتعبير WHILE TEAMS. TEAMNUM = COACHES. TEAMNUM يعرف عمود الوصل ليكون عمود رقم الفريق في كل من الجدولين.

استخدام التكوين «جدول عمود» table. Column يميز أي الأعمدة المشار إليه بالضبط في حالة ما إذا كانت الأعمدة في الجداول المختلفة لها نفس الاسم. ويحدد التعبير* TEAMS. و* SELECT COACHES. إن الوصل عبارة عن وصل تساوي equijoin أي إن كل من أعمدة الجدولين (عمود الوصل مرتين) تظهر في النتيجة. في الواقع يمكننا أن نكتب* SELECT. ببساطة. وكبديل آخر يمكن للوصل أن يشمل كل من أعمدة الجدولين ولكن لجزء فقط من السجلات. مثال ذلك يمكنك أن تصل جدولي COACHES و TEAMS بالنسبة للسجلات التي تشمل DODGERS فقط.

```
SELECT COACHES.*, TEAMS.*
  FROM COACHES, TEAMS
  WHERE TEAMS.TEAMNUM = COACHES.TEAMNUM
    AND TEAMNAME = 'Dodgers'
```

ونتيجة هذا الأمر هي:

TEAMNUM	COACHNAM	COACHPH	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
12	Adams	555-1524	12	Dodgers	Los Angeles	Wilson
12	Baker	555-3690	12	Dodgers	Los Angeles	Wilson

دعنا نبدأ فحص بعض الأمثلة العملية لاستخدام عملية الوصل. «أوجد أسماء المدربين الموجودين في Dodgers» وهذا يشبه ما سبق ذكره وهو أوجد اسم البائع الموجود في حساب عميل معين. والفرق هو أنه هناك بائع واحد فقط لكنه قد يوجد العديد من المدربين وفي هذا المثال تظهر أسماء المدربين في جدول مختلف عن الجدول الموجود به أدله Dodgers وهو نفس نوع المواقف التي تقابلنا في حالة ملفات البائعين والعملاء وبهذا يصبح أمرك ما يلي:

```
SELECT COACHNAM
  FROM COACHES, TEAMS
  WHERE TEAMS.TEAMNUM = COACHES.TEAMNUM
    AND TEAMNAME = 'Dodgers'
```

والذي ينتج عنه ما يلي:

COACHNAM
Adams
Baker

ماذا عن «أسرد جميع المدربين وأرقام الهاتف الخاصة بهم وأسماء الفرق التي يعملون بها»؟ الإجابة هي:

```
SELECT COACHNAM, COACHPH, TEAMNAME
  FROM COACHES, TEAMS
  WHERE TEAMS.TEAMNUM = COACHES.TEAMNUM
```

والنتيجة هي:

COACHNAM	COACHPH	TEAMNAME
Adams	555-1524	Dodgers
Baker	555-3690	Dodgers
Taylor	555-4820	Giants
Jackson	555-2444	Yankees
Victor	555-9327	Tigers
Dooley	555-7321	Tigers

وهنا وصل يشمل ثلاثة جداول. إفرض أنك تطلب سرد أرقام اللاعبين وأسمائهم الذين لعبوا لفريق Dodgers مع وجود عدد سنوات لعب كل منهم لهذا الفريق ومتوسط عدد المضارب لكل منهم. المعلومات الخاصة بتاريخ كل لاعب مع كل فريق توجد في جدول AFFILIATION فقط. هذا معقول حيث أنها تشمل علاقة من العديد إلى العديد وتدخل تكرارات إذا كانت هناك محاولة لدمجها داخل أي من جدولي TEAMS أو PLAYERS. بيانات متوسط السنوات والمضارب التي تصف العلاقة بين لاعب معين وفريق معين توجد كذلك في جدول AFFILIATION. ولكن أسماء اللاعبين فقط تظهر في جدول PLAYERS وإسم الفريق فقط Dodgers يظهر في جدول TEAMS. ماذا يجب أن تفعله في هذه الحالة هو أن تصل جدولي TEAMS و AFFILIATION، مستخدماً عمود TEAMNUM كعمود وصل وتصل جدولي PLAYERS و AFFILIATION مستخدماً عمود PLAYNUM كعمود وصل ويشبه الأمر الذي تصدره ما يلي:

```
SELECT PLAYERS.PLAYNUM, PLAYNAME, YEARS, BATAVG
  FROM PLAYERS, AFFILIATION, TEAMS
  WHERE PLAYERS.PLAYNUM = AFFILIATION.PLAYNUM
    AND AFFILIATION.TEAMNUM = TEAMS.TEAMNUM
    AND TEAMNAME = 'Dodgers'
```

وتكون النتيجة كما يلي:

PLAYNUM	PLAYNAME	YEARS	BATAVG
523	Doe	10	0.257
1779	Jones	1	0.223
5410	Smith	6	0.307
8093	Smith	5	0.250

وكمثال أخير للوصل إفرض أنك أردت أن تجد عمر أكبر اللاعبين سناً في فريق Dodgers. مرة أخرى هذا الطلب يتطلب وصل ثلاثة جداول لتحقيقه. يظهر اسم Dodgers في جدول TEAMS فقط وتظهر بيانات العمر في جدول PLAYERS فقط وتحفظ العلاقة بين هذين الاثنين في جدول AFFILIATION. وعليك أن تصدر الأمر التالي:

```
SELECT MAX(AGE)
  FROM PLAYERS, AFFILIATION, TEAMS
  WHERE PLAYERS.PLAYNUM = AFFILIATION.PLAYNUM
```

```
AND AFFILIATION.TEAMNUM = TEAMS.TEAMNUM
AND TEAMNAME = 'Dodgers'
```

والذي ينتج عنه ما يلي:

MAX(AGE)
32

وهناك تكوين آخر يعرف بأنه إستفساراً جزئياً «Subquery» ويسمح هذا التكوين بعمل تداخل لأوامر SQL بحيث أن الجدول الذي ينتج من أمر SQL يمكن أن يستفسر منه بعد ذلك بواسطة أمر آخر من أوامر SQL. مثال ذلك خذ الاستفسار «أوجد أسماء المدربين وأرقام الهاتف الخاصة بهم والذين يعملون في Dodgers». كما أنت ملم الآن يمكن الإجابة على ذلك باستخدام أمر نوع استفسار جزئي أيضاً. وفي أمر نوع الاستفسار الجزئي التالي تنفذ عبارة SELECT أولاً لأنها داخل القوسين الداخليين جداً. وسوف يتم الحصول بها على رقم الفريق Dodgers ثم بعد ذلك يتم العمل على هذه النتيجة بواسطة أول عبارة SELECT والتي تجد أسماء وأرقام هاتف المدربين الموجودين في هذا الفريق (Dodgers).

```
SELECT COACHNAM, COACHPH
  FROM COACHES
  WHERE TEAMNUM IN
  (SELECT TEAMNUM
      FROM TEAMS
      WHERE TEAMNAME = 'Dodgers')
```

ونتيجة هذا الاستفسار هي:

COACHNAM	COACHPH
Adams	555-1524
Baker	555-3690

عمليات التجديد والحذف والإضافة باستخدام SQL

الثلاثة عمليات لمعالجة البيانات تكون مباشرة فيها عدى السؤال الخاص بتكامل البيانات أو سلامة البيانات. مثال ذلك تذكر أن النظام يوقف إدخالك لسجل في جدول COACHES برقم فريق لا يوجد له سجل في جدول TEAMS ويرجع إلى المستفيد من النظام عملية حفظ بياناته الخاصة متكاملة أو سليمة بهذا المعنى.

أول أوامر معـالجـة البيـانـات هـو UPDATE . إذا تحـرك Dodgers إلى الخلف إلى Brooklyn وتغير المدير إلى Malloy فعليك أن تصـدر ما يـلي لتجديـد جدول TEAMS:

```
UPDATE TEAMS
   SET TEAMCITY = 'Brooklyn',
     TEAMMGR = 'Malloy'
   WHERE TEAMNAME = 'Dodgers'
```

وعند ذلك يشبه جدول TEAMS ما يلي:

TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
12	Dodgers	Brooklyn	Malloy
15	Giants	San Francisco	Johnson
20	Yankees	New York	Simpson
24	Tigers	Detroit	Corbin

يمكن تغيـير العديـد من السجـلات في نفس الـوقت من خـلال مثـل هـذه العمليـة كـذلـك. فـإذا غـيرت شركـة Acme Company من اسمهـا إلى اسم Royal Company فإنك تريد إظهار التغيير في جـدول BATS وعليك أن تصـدر ما يلي:

```
UPDATE BATS
   SET MANUF = 'Royal',
   WHERE MANUF = 'Acme'
```

وعند ذلك يشبه جدول BATS ما يلي:

TEAMNUM	BATNUM	MANUF
12	B01	Royal
12	B09	General
12	B13	Royal
15	B01	Royal
20	B03	United
20	B04	Modern
24	B18	General
24	B21	United

إفرض أن فريق Dodgers قد حل وأنك تريد حذفه من جدول TEAM (مرة أخرى عليك أن تدير نفسك هنا لتعرف ما يحدث للإشارة إلى Dodgers في الجداول الأخرى).

وعليك أن تصدر ما يلي:

```
DELETE
  FROM TEAMS
  WHERE TEAMNAME = 'Dodgers'
```

وبقية جدول TEAMS تأخذ الشكل التالي:

TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
15	Giants	San Francisco	Johnson
20	Yankees	New York	Simpson
24	Tigers	Detroit	Corbin

إذا حدد جزء WHERE قيمة غير فردية فعند ذلك تحذف العديد من السجلات.

لإدخال فريق Condinols في جدول TEAMS عليك أن تصدر ما يلي:

```
INSERT
  INTO TEAMS
  VALUES (17, 'Cardinals', 'St. Louis', 'Gregory')
```

ويشبه جدول TEAMS (بافتراض أنه يظهر كما في شكل 8.5 عند إصدارك هذا الأمر) ما يلي:

TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
12	Dodgers	Los Angeles	Wilson
15	Giants	San Francisco	Johnson
20	Yankees	New York	Simpson
24	Tigers	Detroit	Corbin
17	Cardinals	St. Louis	Gregory

وبينما لا يصبح هذا الجدول مرتباً تذكر أنه إذا كان (كما هو متوقع أن يكون الحال عليه) هناك فهرس على عمود TEAMNUM موجوداً فسوف يدخل النظام بصورة تلقائية القيمة 17 (مع مشير إلى سجل Cardinals في جدول TEAMS) داخل هذا الفهرس في الموقع المناسب لحفظ الترتيب في الفهرس.

الاستفسار بالمثال: Query - by - Example (QBE)

مقدمة:

هناك لغة قاعدة بيانات علاقية أخرى وهي الاستفسار بالمثال - Query - by - Example (QBE) لشركة IBM. وقد كانت هذه هي أولى اللغات العلاقية التي قدمتها شركة IBM تجارياً وإحدى أولى اللغات في الصناعة. بالإضافة إلى ذلك فإن السطح البيني للمستفيد الخاص بها والمرتفع في درجة ابتكاراته يضعها في جانب خاص بالنسبة إلى النظم العلاقية الأخرى.

السطح البيني للمستفيد والمصمم للأفراد التقنيين وغير التقنيين على حد سواء ذو بعدين ويعمل في وسط الخط المفتوح وله إمكانية استفسار موجهة لنهاية طرفية للعرض المرئي. في الواقع كما بدأت تنظر إلى شكل QBE فإنك تلاحظ الفرق بين نهجها المرئي ونهج SQL الخطي والنصي أكثر. (عملياً وجد أن بعض الناس يفضلون أحد النهجين والبعض الآخر يفضلون النهج الآخر). ويبدأ المستفيد بتجديد الجدول (أو الجداول) المطلوب لاستفسار معين. فإذا لم يكن متأكداً من ذلك فيمكن للنظام أن يقدم له قائمة بالجداول المتاحة وأوصافها. وبمجرد اختيار الجدول (أو الجداول) فيعرض النظام تخطيط هذا الجدول مبيناً اسم الجدول وأسماء حقوله. ولإصدار أحد الاستفسارات يعطي المستفيد مثالاً example للمعلومات المطلوبة والتي تكون عبارة عن تحديد اسم متغير في عمود المعلومات المطلوبة. كما يصف المستفيد كذلك أي عوامل تحد من ذلك العمود والأعمدة الأخرى سواء كان ذلك في نفس الجدول أو في ظروف معينة في صندوق خاص للشروط condition box. ولتنفيذ عملية الوصل يحضر المستفيد الجداول المطلوبة على الشاشة في نفس الوقت ويعرف عمود حقل الوصل عن طريق تحديد نفس اسم المتغير تحت أعمدة حقول الوصل المناسبة ويمكن لنتائج الاستفسار أن تعرض على الشاشة أو أن تطبع.

وطبيعة البعدين للاستفسار بالمثال QBE تجعله متسعاً بصفة خاصة للتشغيل الموجه للصيغ. وتشمل السمات الجيدة الأخرى للاستفسار بالمثال QBE ما يلي. ملفات المخرجات (المعروضة) يمكن أن تعالج عن طريق المستفيد في وسط الخط المفتوح. مثال ذلك السجلات الموجودة في أحد عروض المخرجات يمكن أن يختار منها البعض لحذفها قبل أن تستخدم هذه المخرجات في أي شيء آخر. كما أن أسماء الجداول وعناوين الأعمدة يمكن أن تكون مكونة من العديد من الكلمات وكذلك عناوين الأعمدة في الملفات المخزنة يمكن أن تتغير مع تحركها الى جداول المخرجات. كما يمكن إنتاج عناوين جديدة للحقول المحسوبة في المخرجات.

وسوف نصف QBE أكثر في محتوى مثال البيسبول. في الواقع أننا سوف نتبع نفس تسلسل الاستفسارات الذي استخدمناه في وصف SQL. ومع قراءتك خلال هذا القسم فقد تريد أن تشير أو تعود إلى قسم SQL لإجراء مقارنة لكيفية معالجة نفس الاستفسارات بكل من QBE و SQL.

استرجاعات الملف البسيط بواسطة QBE

للبدء دعنا نسترجع محتويات جدول Teams. لعمل ذلك باستخدام QBE يجب أن يكون لديك هيكلاً لجدول الفرق Teams على الشاشة. ويمكن أن تحقق ذلك عن طريق كتابة اسم الجدول في هيكل فارغ ببساطة. فإذا لم تكن تعرف اسم الجدول الذي تريده فإن QBE يقدم لك قائمة بالجداول التي يمكنك الاتصال بها ويمكنك أن تعرفها كلما كنت في حاجة لذلك. وهيكل جدول الفرق Teams يشبه ما يلي:

TEAMS	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR

لاحظ أن أول عنوان عمود ليس حقلاً لكنه يحدد اسم الجدول TEAMS. لعرض محتويات الجدول عليك بأن تدخل الحرف «P» تحت اسم الجدول (كل أوامر QBE تبدأ بحرف أمر يتبعه نقطة):

TEAMS	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
P.				

والنتيجة تكون كما يلي:

TEAMS	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
	12	Dodgers	Los Angeles	Wilson
	15	Giants	San Francisco	Johnson
	20	Yankees	New York	Simpson
	24	Tigers	Detroit	Corbin

لإيجاد السجل الخاص بالفريق رقم 12 فإنك تحدد البحث في الجدول صراحة وذلك بتحديد إنك تريد رقم الفريق هذا فقط:

TEAMS	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
P.	12			

وتصبح النتيجة على ذلك كما يلي:

TEAMS	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
	12	Dodgers	Los Angeles	Wilson

لإسقاط أرقام الفرق وأسمائها فقط فإنك تحدد أمر .P تحت هذه الحقول فقط التي تهمك بدلاً من وصفها تحت اسم الجدول.

TEAMS	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
	P.	P.		

وعلى ذلك فإنك تحصل على ما يلي:

TEAMS	TEAMNUM	TEAMNAME
	12	Dodgers
	15	Giants
	20	Yankees
	24	Tigers

لإيجاد وإعداد اللاعبين وأسمائهم الذين يزيد عمرهم عن 30 سنة فإنك تحدد ما يلي في هيكل جدول اللاعبين:

PLAYERS	PLAYNUM	PLAYNAME	AGE
	P.	P.	>30

وينتج عن ذلك ما يلي:

PLAYERS	PLAYNUM	PLAYNAME
	523	Doe
	8366	Gomez

وبالتحويل إلى جدول أعمال الخبرة WORKEXP يمكنك أن تحصل على قائمة بأسماء المدربين الذين عملوا مدربين في الكليات لأكثر من خمسة سنوات وذلك بتحديد ما يلي:

WORKEXP	TEAMNUM	COACHNAM	EXPTYPE	EXPYRS
	P.	P.	College Coach	>5

وينتج عن ذلك ما يلي:

WORKEXP	TEAMNUM	COACHNAM
	12	Adams
	15	Taylor

ولإيجاد الفرق التي تستخدم مضارب من صنع Acme Company. فإنك تحدد ما يلي:

BATS	TEAMNUM	BATNUM	MANUF
	P.		Acme

وحيث أن QBE يحذف الصفوف المزدوجة الموجودة في النتيجة تلقائياً فنتيجة هذا الاستفسار تكون كما يلي:

BATS	TEAMNUM
	12
	15

أما إذا أردت أن تحصل على كل السجلات التي تشمل Acme Company مرة أخرى مع رقم الفريق فقط في المخرجات، وبالرغم من أن ذلك لا يعطي كل المعنى الموجود في المثال، فعليك أن تحدد ما يلي:

BATS	TEAMNUM	BATNUM	MANUF
	P.ALL.		Acme

لتحصل على ما يلي:

BATS	TEAMNUM
	12
	12
	15

هل تذكر استفسار SQL الموجود في هذا الفصل والذي يشمل إيجاد اللاعبين الذين تقع أعمارهم بين 25 و 29 سنة مع ترتيب النتيجة؟ حيث أن كل من هاتين المواصفتين تشمل العمر وكتابته تحت عمود AGE في هيكل Players وتكون معقدة لمعالجتها بطريقة QBE فيجب أن تعد جزءاً من الاستفسار في صورة صندوق شرط condition box.

PLAYERS	PLAYNUM	PLAYNAME	AGE
P.			AO._OLD

CONDITIONS
25 <= _OLD <= 29

لاحظ أن أمر «P» قد عاد إلى موقعه تحت اسم الجدول لأن النتيجة يجب أن تشمل كل أعمدة الجدول. ويحدد «AO»* الترتيب التصاعدي للصفوف الناتجة طبقاً للعمر. والمواصفة في صندوق CONDITIONS متصلة بعمود AGE بواسطة المتغير المشترك والاختياري «OLD -» والنتيجة هي ما يلي:

PLAYERS	PLAYNUM	PLAYNAME	AGE
	1779	Jones	25
	4280	Cohen	25
	2007	Dobbs	27
	5410	Smith	27
	1131	Johnson	28

ولإنتاج قائمة بالمضارب التي تنتجها شركة General Company أو شركة United Company عليك أن تحدد ما يلي:

BATS	TEAMNUM	BATNUM	MANUF
P.			_M

CONDITIONS
_M = ('General' | 'United')

والذي ينتج عنه ما يلي:

BATS	TEAMNUM	BATNUM	MANUF
	12	B09	General
	20	B03	United
	24	B18	General
	24	B21	United

وصيغة QBE الخاصة باسترجاع كل حدوثات أحد الحقول الذي له جزء مشترك تتطلب توضيح الجزء المشترك كثابت والجزء غير المعروف كمتغير اختياري. ولإيجاد سجلات كل هؤلاء اللاعبين الذين تبدأ أسماء العائلة لهم بالحرف S عليك أن تحدد ما يلي:

PLAYERS	PLAYNUM	PLAYNAME	AGE
P.		S_NAME	

وينتج عن ذلك ما يلي:

PLAYERS	PLAYNUM	PLAYNAME	AGE
	358	Stevens	21
	5410	Smith	27
	8093	Smith	21

وتوجد أوامر في QBE تكافيء الداول المبنية داخلياً التي سبق مناقشتها في SQL. مثال ذلك لإيجاد عدد السجلات الموجودة في جدول الفرق Teams فإنك تحدد ما يلي:

TEAMS	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
	P.CNT.ALL.			

وتكون النتيجة كما يلي:

TEAMS	TEAMNUM CNT
	4

لإيجاد عدد المنتجين المختلفين الذي يقومون بتوريد المضارب للفرق فإنك تستخدم الكلمة الرئيسة UNQ وهي اختصار فريد Unique وهي كلمة في QRE تناظر كلمة DISTINCT في SQL:

BATS	TEAMNUM	BATNUM	MANUF
			P.CNT.UNQ.ALL.

وينتج عن ذلك ما يلي:

BATS	MANUF CNT
	4

وبالعودة إلى صندوق الشرط يمكنك أن تحصل على عد لعدد اللاعبين الذين يزيد عمرهم عن 30 سنة وذلك بتحديد ما يلي:

PLAYERS	PLAYNUM	PLAYNAME	AGE
			P.CNT.ALL._X

CONDITIONS
_X > 30

تذكر أن المتغير «X -» هو متغير اختياري وتكون النتيجة كما يلي:

PLAYERS	AGE CNT
	2

ولإيجاد إجمالي عدد سنوات خبرة المدرب Adams في Dodgers (الفريق رقم 12) فيمكنك استخدام جدول أعمال الخبرة Work Experience وتحدد ما يلي:

WORKEXP	TEAMNUM	COACHNAM	EXPTYPE	EXPYRS
	12	Adams		P.SUM.ALL.

تكون النتيجة كما يلي:

WORKEXP	EXPYRS SUM
	15

ومكافيء مواصفة التجميع بواسطة «Group By» في SQL هو الكلمة الرئيسية «G» في QBE ولإيجاد إجمالي عدد سنوات خبرة كل مدرب فإنك تكتب ما يلي:

WORKEXP	TEAMNUM	COACHNAM	EXPTYPE	EXPYRS
	P.G.	P.G.		P.SUM.ALL.

والذي ينتج عنه ما يلي:

WORKEXP	TEAMNUM	COACHNAM	EXPYRS SUM
	12	Adams	15
	12	Baker	9
	15	Taylor	15
	24	Victor	12

استرجاعات ملفات متعددة بواسطة QBE (الوصلات):

لتحديد أمر يتطلب وصل في QBE فيجب أن تعرض كل الجداول المشمولة في نفس الوقت. وحقول الوصل تحدد بواسطة وضع نفس اسم المتغير الاختياري تحت أعمدة وصلة المواءمة في الجداول المختلفة. فإذا احتوت المخرجات على حقول من أكثر من جدول واحد فيجب عند ذلك انتاج هيكل إضافي يصف شكل المخرجات. وكمثال أولى كما هو في حالة SQL فيما يلي وصله بين جدولي COACHES و TEAMS:

COACHES	TEAMNUM	COACHNAM	COACHPH
	_TEAM	_COACH	_PHONE

TEAMS	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
	_TEAM	_TEAMNAM	_CITY	_MGR

	TEAMNUM	COACHNAM	COACHPH	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
P.	_TEAM	_COACH	_PHONE	_TEAM	_TEAMNAM	_CITY	_MGR

لاحظ الطريقة التي تحدد بها أسماء المتغيرات الاختيارية في جدولي COACHES و TEAMS والتي توجه النظام في تشييد المخرجات طبقاً لوضعها في هيكل المخرجات. المتغير TEAM - يعد حقلي الوصل ويحمل نتيجتهما إلى جدول المخرجات كذلك. وتكون النتيجة كما يلي:

TEAMNUM	COACHNAM	COACHPH	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
12	Adams	555-1524	12	Dodgers	Los Angeles	Wilson
12	Baker	555-3690	12	Dodgers	Los Angeles	Wilson
15	Taylor	555-4820	15	Giants	San Francisco	Johnson
20	Jackson	555-2444	20	Yankees	New York	Simpson
24	Victor	555-9327	24	Tigers	Detroit	Corbin
24	Dooley	555-7321	24	Tigers	Detroit	Corbin

لإيجاد بيانات المدرب والفريق للمدربين الموجودين في Dodgers فقط عليك بأن تحدد ما يلي:

COACHES	TEAMNUM	COACHNAM	COACHPH
	_TEAM	_COACH	_PHONE

TEAMS	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
	_TEAM	_TEAMNAM	_CITY	_MGR

	TEAMNUM	COACHNAM	COACHPH	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
P.	_TEAM	_COACH	_PHONE	_TEAM	_TEAMNAM	_CITY	_MGR

CONDITIONS
_TEAMNAM = Dodgers

وتكون النتيجة كما يلي:

TEAMNUM	COACHNAM	COACHPH	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
12	Adams	555-1524	12	Dodgers	Los Angeles	Wilson
12	Baker	555-3690	12	Dodgers	Los Angeles	Wilson

وبالاستمرار باتباع نمط الاستفسارات الموجود مع SQL فإن الاستفسار التالي يكون «أوجد أسماء مدربي Dodgers». تذكر أن هذا يتطلب توصيل لأن أسماء المدربين تظهر في جدول مختلف عن الجدول الذي يشير إلى Dodgers وعليك أن تحدد ما يلي:

COACHES	TEAMNUM	COACHNAM	COACHPH
	_TEAM	P.	

TEAMS	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
	_TEAM	Dodgers		

وحيث أن المخرجات وهي اسم المدرب تأتي من أحد الجداول المشمولة فقط فإننا نعفي من هيكل المخرجات الإضافي الذي كنا في حاجة إليه قبل ذلك. مرة أخرى يحدد المتغير TEAM -. حقول الوصل ويحقق الثابت Dodgers حدود الاستفسار. والنتيجة تصبح كما يلي:

COACHES	COACHNAM
	Adams Baker

يلي ذلك «أسرد المدربين وأرقام الهاتف لهم وأسماء الفرق التي يعملون لها» وعليك أن تحدد لذلك ما يلي:

COACHES	TEAMNUM	COACHNAM	COACHPH
	_TEAM	_COACH	_PHONE

TEAMS	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
	_TEAM	_TNAME		

	COACHNAM	COACHPH	TEAMNAME
P.	_COACH	_PHONE	_TNAME

لاحظ العودة إلى شكل هيكل المخرجات وذلك لوجود حقول من كل من الجدولين في المخرجات والتي تشبه ما يلي:

COACHNAM	COACHPH	TEAMNAME
Taylor	555-4820	Giants
Adams	555-1524	Dodgers
Baker	555-3690	Dodgers
Victor	555-9327	Tigers
Dooley	555-7321	Tigers
Jackson	555-2444	Yankees

والآن إلى توصيل ثلاثة جداول. يوجه الاستفسار لرد أرقام اللاعبين وأسمائهم الذين لعبوا لفريق Dodgers مع وجود عدد سنوات لعب كل منهم ومتوسط مضاربهم. الجداول المطلوبة هي PLAYERS و TEAMS و AFFILIATION. وتظهر معلومات اللاعب في جدول PLAYERS والإشارة إلى Dodgers تكون في جدول TEAMS والاتصال بينهما يكون في جدول AFFILIATION:

PLAYERS	PLAYNUM	PLAYNAME	AGE
	_PLAYN	_PNAME	

AFFILIATION	PLAYNUM	TEAMNUM	YEARS	BATAVG
	_PLAYN	_TNUM	_YRS	_BAVG

TEAMS	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
	_TNUM	Dodgers		

	PLAYNUM	PLAYNAME	YEARS	BATAVG
P.	_PLAYN	_PNAME	_YRS	_BAVG

يربط PLAYN - جدولي PLAYERS و AFFILIATION معاً ويربط TNUM - جدولى AFFILIATION و TEAMS معاً. وتكون النتيجة كما يلي:

PLAYNUM	PLAYNAME	YEARS	BATAVG
523	Doe	10	0.257
1779	Jones	1	0.223
5410	Smith	6	0.307
8093	Smith	5	0.250

وآخر مثال وصل يدعى لعمر أكبر لاعبي Dodgers سناً. وهنا مرة أخرى تكون معلومات العمر في جدول PLAYERS وتكون الإشارة الى Dodgers في جدول TEAMS والاتصال بينهما يكون في جدول AFFILIATION.

PLAYERS	PLAYNUM	PLAYNAME	AGE
	_PLAYN		P.MAX.ALL.

AFFILIATION	PLAYNUM	TEAMNUM	YEARS	BATAVG
	_PLAYN	_TNUM		

TEAMS	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
	_TNUM	Dodgers		

وحيث أن النتيجة تحتوي على العمر فقط. فلا يكون هناك حاجة إلى هيكل للنتيجة وتكون النتيجة كما يلي:

PLAYERS	AGE MAX
	32

عمليات التجديد والحذف والإضافة باستخدام QBE:

عمليات معالجة البيانات باستخدام QBE مباشرة، وتتبع نفس التكوين العام الذي سبق لنا العمل به. لتجديد جدول TEAMS ليعكس الانتقال إلى Brooklyn وتعيين Malloy كمدير فإنك تحدد ما يلي:

TEAMS	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
		Dodgers	U. Brooklyn	U.Malloy

ومن الواضح أن «U» هو أمر التجديد. أما إذا ما حددنا الآن أمر .P في هيكل TEAMS فإننا نرى أنه يشبه ما يلي:

TEAMS	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
	12	Dodgers	Brooklyn	Malloy
	15	Giants	San Francisco	Johnson
	20	Yankees	New York	Simpson
	24	Tigers	Detroit	Corbin

ولتغيير اسم Acme Company إلى Royal Compan في كل سجل من سجلات Acme في جدول BATS فهناك صيغتان متكافئتان للأمر. الصيغة الأولى هي:

BATS	TEAMNUM	BATNUM	MANUF
			Acme U. Royal

والصيغة الثانية هي:

BATS	TEAMNUM	BATNUM	MANUF
	_T	_B	Acme
U.	_T	_B	Royal

وفي الصيغة الثانية تستخدم المتغيرات T - و B -. و TEAMNUMوBATNUM مع بعضهما يكونان المفتاح الفردي للجدول. ويوجد السطر الأول السجلات التي يوجد فيها Acme. أما السطر الثاني مع T - فينقل قيم الحقل الرئيسي مع وضعها لـ Royal. وفي أي من الحالتين يوضح أمر P. إن جدول BATS يبين ما يلي:

BATS	TEAMNUM	BATNUM	MANUF
	12	B01	Royal
	12	B09	General
	12	B13	Royal
	15	B01	Royal
	20	B03	United
	20	B04	Modern
	24	B18	General
	24	B21	United

ولحذف سجل Dodgers فإنك تحدد ما يلي:

TEAMS	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
D.		Dodgers		

وبالطبع فإن .D هو أمر الحذف. ويبين TEAMS الآن ما يلي:

TEAMS	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
	15	Giants	San Francisco	Johnson
	20	Yankees	New York	Simpson
	24	Tigers	Detroit	Corbin

وأخيراً فإنك تستخدم أمر .I لإدخال سجل لـ Cardinals داخل جدول TEAMS الأصلي:

TEAMS	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
I.	17	Cardinals	St. Louis	Gregory

وعرض جدول TEAMS يظهر منه ما يلي:

TEAMS	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
	12	Dodgers	Brooklyn	Malloy
	15	Giants	San Francisco	Johnson
	20	Yankees	New York	Simpson
	24	Tigers	Detroit	Corbin
	17	Cardinals	St. Louis	Gregory

بينما الوضع في نهاية الجدول يعكس الموقع الواقعي فيمكن أن ترتب أي مخرجات مستقبلية طبقاً لأي حقل بأوامر التسلسل التصاعدي أو التنازلي المناسبة.

مراجع: References

Cheng, J.M., Loosley, C.R., Shibamiya, A., and Worthington, P.S., «IBM Database 2 Performance: Design, Implementation, and Tuning», IBM Systems Journal, vol. 23, no. 2, 1984, pp. 189 - 210.

Date, C.J., A Guide to DB2, Addison - Wesley, Reading, MA, 1984.

Date, C.J., An Introduction to Database Systems, 3rd., vol. 1, Addison - Wesley, Reading, MA, 1981.

Haderle, D.J., and Jackson, R.D., «IBM Database 2 Overview», IBM Systems Journal, vol. 23, no. 2 1984, pp. 112 - 125.

IBM Corporation, IBM DATABSE 2 Data Base Planning and Administration Guide, IBM From No. SC26 - 4077, 1983.

IBM Corporation, IBM DATABSE 2 Relational Concepts, IBM From No. GG24 - 1581, 1983.

IBM Corporation, IBM DATABSE 2 SQL Usage Guide, IBM From No. GG24 - 1583, 1983.

IBM Corporation, Query - by - Example Program Description /Operations Manual, IBM From No. SH20 - 2077, 1980.

IBM Corporation, Query - by - Example Terminal User's Guide, IBM From No. SH20 - 2078, 1983.

IBM Corporation, Query - by - Example Terminal User's Guide - VMISP, IBM From No. SH24 - 5043, 1983.

IBM Corporation, SQL/Data System Ternimal User's Guide - VMISP, IBM From No. SH24 - 5045, 1983.

Kim, W., «Relational Database Systems», ACM Computing Surveys, Vol. 11, no. 3, September, 1979, pp. 185 - 212.

King, J.M., Evaluating Data Base Management Systems, Van Nostrand Reinhold, New York, 1981.

Kroenke, D., Database Processing, 2nd ed., Science Research Associates, Chicago, IL, 1983.

أسئلة وتمارين: Questions and Exercises

١ ـ ما العلاقة ؟ ما المفاتيح الأولية والمرشحة والأجنبية ؟

٢ ـ ما التمييز بين قاعدة البيانات العلاقية وأحد الفئات المعتادة من الملفات الخطية المستقلة ؟

٣ ـ كيف يمكنك أن تقارن بين نظام إدارة قاعدة بيانات علاقية مع فئة من برامج التطبيقات التي تقوم بتشغيل فئة من الملفات الخطية المعتادة؟

٤ ـ كيف يمكنك أن تقارن فئة من برامج التطبيقات التي تقوم بتشغيل بيانات داخل بيئة نظام لإدارة قاعدة بيانات علاقية مع فئة من برامج التطبيقات التي تقوم بتشغيل فئة من الملفات الخطية؟

٥ ـ صف بكلمات من عندك الضرورة اللازمة لعمل نظام إدارة قاعدة بيانات علاقية.

٦ ـ صف عملية الاختيار. ماذا تحققه هذه العملية؟

٧ ـ صف عملية الإسقاط. ماذا تحققه هذه العملية؟

٨ ـ بالإشارة إلى علاقة العملاء الموجودة في شكل 8.3b وباستخدام أسلوب الأوامر العلاقية الذي سبق توضيحه في هذا الفصل حدد نتائج الأوامر التالية:

أ ـ اختيار صفوف من علاقة العملاء التي يكون فيها رقم العميل CUS-TOMER NUMBER = 1047

ب ـ اختيار صفوف من علاقة العملاء التي يكون فيها SALESPERSON

NUMBER = 361

جـ ـ إسقاط CUSTOMER NUMBER و CITY على علاقة العملاء.

د ـ إسقاط CITY على علاقة العملاء.

هـ ـ اختيار صفوف من علاقة العملاء التي يكون فيها SALESPERSON NUMBER = 137 و CITY على هذه النتيجة.

٩ ـ صف عملية الوصل. ماذا تحققه هذه العملية؟

١٠ ـ اعتبر أول ثلاثة صفوف فقط من كل من علاقة العملاء وعلاقة البائعين الموجودة في شكل 8.3. استخدم اسلوب الأوامر العلاقية الذي سبق توضيحه في هذا الفصل في تحديد نتائج تسلسل الأوامر التالية:

أ ـ توصيل علاقة البائعين مع علاقة العملاء مستخدماً SALESPERSON NUMBER الموجود في كل منها كحقل توصيل. واختيار صفوف من النتيجة يكون فيها SALESPERSON NUMBER = 137 واسقاط SALESPERSON NUMBER و CUSTOMER NUMBER و CITY على هذه النتيجة.

ب ـ توصيل علاقة البائعين وعلاقة العملاء مستخدماً SALESPERSON NUMBER الموجود في كل منها كحقل توصيل. واختيار صفوف من هذه النتيجة يكون فيها CITY = Harfard. وإسقاط SALESPERSON NAME على هذه النتيجة.

١١ ـ اعتبر علاقة البائعين وعلاقة العملاء الموجودة في شكل 8.3 باستخدام أسلوب الأوامر العلاقية الذي سبق توضيحه في هذا الفصل اكتب عبارات علاقية للإجابة على الأسئلة التالية:

أ ـ إيجاد سجل البائع Carlyle.

ب ـ إيجاد سجلات كل العملاء الموجودين في بوستن Bosten.

جـ ـ إعداد قائمة بكل العُملاء (طبقاً لرقم العميل) مع المدينة الموجودة بها كل عميل.

د ـ إعداد قائمة بالعملاء (طبقاً لرقم العميل والمدينة) المسؤول عنهم البائع رقم 186.

هـ ـ إعداد قائمة بالعملاء (طبقاً لرقم العميل والمدينة) المسؤول عنهم البائع Adams.

و ـ ما أسماء البائعين المسؤولين عن الحسابات الموجودة في نيويورك ؟

١٢ ـصف بكلمات من عندك بالمقارنة بين طرق تحقيق تكامل البيانات (مفهوم الوصل) في نظم إدارة قواعد البيانات العلاقية من ناحية ونظم إدارة قواعد البيانات الهرمية والشبكية من ناحية أخرى.

١٣ ـصف بكلمات من عندك المقارنة بين المرونة في التصميم والتعديل المقدم بواسطة نظم إدارة قواعد البيانات العلاقية من ناحية ونظيرة لنظم إدارة قواعد البيانات الهرمية والشبكية من ناحية أخرى.

١٤ ـما هي العوامل الرئيسية التي تؤثر على إدارة نظم إدارة قواعد البيانات العلاقية ونظم إدارة قواعد البيانات الهرمية والشبكية؟

١٥ ـ ما التكامل الإشاري referential integrity؟ ناقشه بالنسبة إلى نظم إدارة قواعد البيانات العلاقية والهرمية والشبكية.

١٦ ـإعمل تخطيطاً للمفاهيم الموجودة وراء القائم بعمل الأمثلية Optimizer في نظام SQL لشركة IBM هل هناك طرق أخرى يمكنك التفكير فيها لتحسين أداء قاعدة البيانات العلاقية؟

فيما يلي أربع علاقات تشتمل على قاعدة بيانات لإصلاح الأجهزة بإحدى الشركات والتي سوف تستخدم في كل الأسئلة التالية عن SQL و QBE. تحتفظ الشركة ببيانات عن التقنيين بها (اسم العامل ورقم ولقبه) وأنواع الأجهزة التي تؤدي لها الخدمة مع عدد الساعات اللازمة لإصلاح كل منها والأجهزة المحددة (طبقاً لرقم التسلسل) التي حددت لها عقود صيانة وأي التقنيين معد لأداء الخدمة لأي أنواع الأجهزة (بما في ذلك عدد سنوات خبرة هذا التقني في هذا النوع الخاص من الأجهزة).

TECHNUM	TECHNAME	TITLE
062	Regan	Senior
297	Mercer	Trainee
553	Hewitt	Senior
718	Stein	Staff

TECHNICIANS Relation

APPLTYPE	CATEGORY	RATE
Washer	Major	20
Dryer	Major	20
Toaster	Minor	10
Blender	Minor	8
Freezer	Major	25

TYPES Relation

APPLNUM	APPLTYPE	AGE	OWNER
00730	Toaster	5	M. Peters
08331	Freezer	7	R. Carr
16365	Freezer	3	S. Small
22770	Blender	1	C. Ramos
40034	Washer	10	S. Small
47222	Dryer	12	S. Small
49371	Freezer	9	N. Rice
58449	Toaster	7	L. Shaw
60558	Washer	3	M. Peters
72598	Washer	7	S. Bauer

APPLIANCES Relation

TECHNUM	APPLTYPE	YRSQUAL
553	Dryer	15
062	Washer	18
297	Toaster	1
297	Dryer	1
718	Washer	5
553	Freezer	12
062	Blender	14
062	Freezer	10
062	Dryer	12

QUALIFICATION Relation

١٧ ـباستخدام قاعدة بيانات شركة صيانة الأجهزة السابقة حول أوامر SQL التالية إلى اللغة الإنجليزية (أو إلى اللغة العربية) مع إظهار نتائجها بالبيانات المبينة.

```
a. SELECT *
     FROM TECHNICIANS
     WHERE TITLE = 'Senior'
b. SELECT APPLNUM, OWNER, AGE
     FROM APPLIANCES
     WHERE APPLTYPE = 'Freezer'
     ORDER BY AGE
c. SELECT APPLTYPE, OWNER
     FROM APPLIANCES
     WHERE AGE BETWEEN 5 AND 10
d. SELECT COUNT(*)
     FROM TECHNICIANS
e. SELECT AVG(RATE)
     FROM TYPES
     WHERE CATEGORY = 'Major'
f. SELECT CATEGORY, AVG(RATE)
     FROM TYPES
     GROUP BY CATEGORY
g. SELECT APPLNUM, OWNER
     FROM TYPES, APPLIANCES
     WHERE TYPES.APPLTYPE = APPLIANCES.APPLTYPE
       AND CATEGORY = 'Minor'
h. SELECT APPLTYPE, MAX(YRSQUAL)
     FROM TECHNICIANS, QUALIFICATION
     WHERE TECHNICIANS.TECHNUM = QUALIFICA-
         TION.TECHNUM
         AND TECHNAME = 'Regan'
i. SELECT APPLNUM, OWNER
     FROM TECHNICIANS, QUALIFICATION, APPLIANCES
     WHERE TECHNICIANS.TECHNUM = QUALIFICA-
         TION.TECHNUM
         AND QUALIFICATION.APPLTYPE = APPLIANCES.
         APPLTYPE
         AND TECHNAME = 'Hewitt'
```

١٨ ـاكتب أوامر SQL للإجابة على الطلبات التالية من قاعدة بيانات شركة إصلاح الأجهزة. (قد تكون بعض الاستفسارات مهمة بالنسبة للحقول

المطلوبة في الاستجابة لها بالضبط استخدم أحكامك الشخصية).

أ ـ ما الأجهزة الرئيسية ؟

ب ـ من الذين يمتلكون أجهزة فريزر؟

جـ ـ ما هي الأرقام المتسلسلة وما هي أعمار مسخنات الخبز toasters المحرر لها عقود خدمة؟

د ـ ما هو اسم التقني الأكثر تأهيلاً لإصلاح أجهزة الفريزر؟

هـ ـ ما هو متوسط عمر الغسالات الموجود لها عقود خدمة؟

و ـ أي المنازل يجب أن يزورها Stein؟

ر ـ ما هو عدد ألقاب الأعمال الممثل في قاعدة البيانات؟

ز ـ ما هو متوسط عمر كل جهاز رئيس لكل مالك؟

س ـ من هم الذين زاد عمر أجهزة الفريزر الخاصة بهم من ستة أعوام؟

ش ـ ما هي معدلات الفواتير المختلفة وعلى أي الأجهزة هذه المعدلات بالنسبة للمالك S. Small؟

ع ـ ما هو متوسط معدل الفاتورة للأجهزة الرئيسية المؤهل لتصليحها Regan؟

١٩ ـ ما هو شعورك بالنسبة للفرق بين للسطح البيني ذو البعدين لـ QBE بالمقارنة مع السطح البيني لـ SQL؟

٢٠ ـ باستخدام قاعدة بيانات شركة إصلاح الأجهزة حول أوامر QBE التالية إلى اللغة الإنجليزية (أو اللغة العربية) محدداً نتائجها بالبيانات المبينة.

a.

TYPES	APPLTYPE	CATEGORY	RATE
	P.		>18

b.

APPLIANCES	APPLNUM	APPLTYPE	AGE	OWNER
P.		Freezer	<6	

c.

APPLIANCES	APPLNUM	APPLTYPE	AGE	OWNER
	P.	_APPL		P.

CONDITIONS
_APPL = ('Washer' | 'Dryer')

d.

TECHNICIANS	TECHNUM	TECHNAME	TITLE
			P.CNT.ALL._QQ

CONDITIONS
_QQ = 'Senior'

e.

APPLIANCES	APPLNUM	APPLTYPE	AGE	OWNER
			P.AVG.ALL.	P.G.

f.

TECHNICIANS	TECHNUM	TECHNAME	TITLE
	_NUM	P.	

QUALIFICATION	TECHNUM	APPLTYPE	YRSQUAL
	_NUM	Dryer	>10

g.

TYPES	APPLTYPE	CATEGORY	RATE
	_APPL	_CAT	

APPLIANCES	APPLNUM	APPLTYPE	AGE	OWNER
	_NUM	_APPL	>9	_OWN

	TYPE	CATEGORY	OWNER	NUMBER
P.	_APPL	_CAT	_OWN	_NUM

h.

TECHNICIANS	TECHNUM	TECHNAME	TITLE
	_NUM	P.	

QUALIFICATION	TECHNUM	APPLTYPE	YRSQUAL
	_NUM	_APPL	

TYPES	APPLTYPE	CATEGORY	RATE
	_APPL		>19

٢١ ـ اكتب أوامر QBE للإجابة على الطلبات التالية من قاعدة بيانات شركة إصلاح الأجهزة. (قد تكون بعض الاستفسارات مهمة بالنسبة للحقول المطلوبة في الاستجابة لها بالضبط ـ استخدم أحكامك الشخصية).

أ ـ ما هي الأجهزة الرئيسية؟

ب ـ من الذين يمتلكون أجهزة فريزر؟

جـ ـ ما هي الأرقام المتسلسلة وما هي أعمار سخنات الخبز المحرر لها عقود خدمة؟

د ـ ما هو اسم التقني الأكثر تأهيلاً لإصلاح أجهزة الفريزر؟

هـ ـ ما هو متوسط عمر الغسالات الموجود لها عقود خدمة؟

و ـ أي المنازل يجب أن يزورها Stein؟

ر ـ ما هو عدد ألقاب الأعمال الممثل في قاعدة البيانات؟

ز ـ ما هو متوسط عمر كل جهاز رئيسي لكل مالك؟

س ـ منهم الذين زاد عمر أجهزة الفريز الخاصة بهم عن ستة أعوام؟

ش ـ ما هي معدلات الفواتير المختلفة وعلى أي أجهزة هذه المعدلات بالنسبة إلى المالك S. Small؟

ع ـ ما هو متوسط معدل الفاتورة للأجهزة الرئيسية المؤهل لتصليحها Regan؟

الفصل التاسع
المنهج شبه العلاقي لقاعدة البيانات
The Pseudo - Relational Approach to Database

مقدمة : Introduction

العديد من العاملين في مجال قواعد البيانات يعتبرون نماذج تصميم نظم قواعد البيانات الهرمية والشبكية والعلاقية هي النماذج الوحيدة الموجودة حالياً. وسوف نتحدث عن منهج رابع لقواعد البيانات والذي يسمى في المراجع بالمنهج شبه العلاقي pseudo - relational أو بتكامل الملف المسطح flat - file integrated أو نصف العلاقي semi - relational أو شبه العلاقي relational like.

والمنهج شبه العلاقي هو هجين في واقع الأمر يتكون من العديد من سمات مناهج قواعد البيانات التي سبق ذكرها بالفعل في هذا الكتاب. وتكوين البيانات الأساسي هو ملفات خطية بسيطة وهذه سمة موجودة في النظم العلاقية. ويتطلب هذا المنهج مثل ما هو الحال في المنهج الهرمي أو الشبكي بعض صيغ الاتصال بين السجلات المرتبطة ببعضها والموجودة في ملفات مختلفة والتي يجب أن تعد مسبقاً بعمل الاستفسار في وسط خط مفتوح وتخزن مع قاعدة البيانات.

ولا يوجد تعريف رسمي للمنهج شبه العلاقي لإعداد نظام إدارة قواعد البيانات. ووصف المنهج شبه العلاقي هو في واقع الأمر شيء من المفاهيم الأساسية، مثل الذي سبق ذكره مرتبطاً بالاختلافات في تنفيذ مفهوم أن مطوري نظم قواعد البيانات المختلفة اسهموا فعلاً في تصميات هذا المنهج.وهناك العديد من مثل هذه الصيغ المختلفة. وسوف نجري مقارنات أكثر بين المنهج شبه العلاقي والمناهج الأخرى مع التقدم في هذا الفصل. وكلمة جانبية هي أن الشخص يرى إصطلاح تكوين الملف المعكوس inverted file structure

مستخدماً في وصف صيغة تخزين البيانات المحتوية على ملفات خطية بسيطة لها فهارس مبنية على كل أو معظم حقولها. وسواء تؤهل النظم المبنية على مثل هذه التكوينات كنظم قواعد بيانات أم لا فذلك يعتمد أساسا،كما يجب أن يكون ذلك واضحاً لك الآن على ما إذا كان لها نفس نوع إمكانية الوصل أم لا.

مفاهيم قواعد البيانات شبه العلاقية : Pseudo - Relational Database Concepts

شكل 9.1 عبارة عن تكرار لشكل 8.3 وهو ملفات البائعين والعملاء (سوف نسقط إصطلاح «علاقة» من هذا الفصل) الذي استخدمناه في الفصل السابق. لاحظ أننا أضفنا أرقام سجلات نسبية على اليسار والتي تحدد ببساطة أي السجلات الأول وأيها الثاني وأيها الثالث وهكذا في الملف.

مرة أخرى لقد واجهنا المشكلة المعتادة الخاصة بطلب حفظ الطبيعة غير

	SALESPERSON NUMBER	SALESPERSON NAME	COMMISSION PERCENTAGE	YEAR OF HIRE
1	137	Baker	10	1975
2	186	Adams	15	1971
3	204	Dickens	10	1963
4	361	Carlyle	20	1963

(a)

	CUSTOMER NUMBER	SALESPERSON NUMBER	CITY
1	0121	137	New York
2	0839	186	Hartford
3	0933	137	Boston
4	1047	137	Boston
5	1525	361	Newark
6	1700	361	Washington
7	1826	137	New York
8	2198	204	New York
9	2267	186	New York

(b)

شكل 9.1

(أ) ملف البائعين (ب) ملف العملاء

المتكررة للبيانات بينها يحدث تكامل في نفس الوقت بين الملفين. لقد سبق أن رأينا كيفية تكوين البيانات وتطوير تكاملها بالمشيرات الطبيعة داخل هذه التكـوينات في المنهج الهرمي والمنهج الشبكي. كما سبق أن رأينا كذلك كيفية تكـامل البيـانات في المنهج العلاقي وتخزينها في علاقات طائرة عنـد وقت التنفيذ خـلال حقول الـوصل. دعنا نحاول شيئاً آخراً بالنسبة للمنهج شبه العلاقي.

يبـين شكل 9.2 ملف وصـل link - file يحـدد أي السجـلات من ملفـات البائعين والعملاء مرتبطة مع بعضها. بالطبع فإن طريقة ارتباط السجـلات تتماشى مع مفهوم الوصل. مثـال ذلك السجـل رقم 1 من ملف البائعـين به القيمـة 1.37 لحقل SALESPERSON NUMBER كذلك. وبالمناسبة مع مفهوم الوصل فـإن السجل يتم انتاجه في ملف الوصـل في شكل 9.2 (أول سجـل في ملف الوصـل)

والذي يحدد أن السجل رقم 1 للملف البائعين والسجل رقم 1 للملف العملاء مرتبطين ببعضهـما. في واقع الأمـر يحتوي ملف الـوصل هـذا عـلى سجـل لكـل خليط من سجلات ملفى العملاء والبائعين التي لهـا نفس القيمة لحقـل SALESPERSON NUMBER. أليس هذا هو الإحساس بما سبق أن وصفناه بأنه وصل؟ بـالتأكيـد أنه ذلك الوصل.

وتبقى البيانات في المنهـج شبه العـلاقي كملفات خـطية بسيـطة مستقلة مع انتاج ملفات وصل إضافية بصورة طبيعية تسبق حدوث أي استفسـار يدخـل على البيـانات والتي تحـدد كيفية ارتبـاط سجلات من ملفـات مختلفة مـع بعضها بنفس إحساس مفهوم الوصل.

SALESPERSON RECORD	CUSTOMER RECORD
1	1
1	3
1	4
1	7
2	2
2	9
3	8
4	5
4	6

شكل 9.2
ملف الوصل بين ملفى البائعين والعملاء

وتتوازى مميزات وعيوب المنهج شبه العلاقي مع أوجه مفاهيم قواعد البيانات الأخرى والتي استخلص منها هذا المنهج . فللمنهج شبه العلاقي الكثير مما يعتبر بساطة نسبية للملفات الخطية البسيطة كما أنه له رفعاً لأداء الوصلات التي تحدث مسبقاً مع تخزين نتائج هذه الوصلات لتستخدمها الاستفسارات في المستقبل. وفي الناحية الأخرى فإنه به العيب بأنه لا يوجد له علاقة حقيقية طائرة on-the-fly لإمكانية الوصل. كما أنه في الأوقات التي تتغير فيها البيانات في الملف يجب أن يأخذ النظام وقتاً لتحديث أي فهارس تتأثر بالتغيير . فإذا تم تأجيل تحديث هذه الفهارس فيجب أن يؤدي النظام تشغيلاً إضافياً لعمل بحث مفهرس للسجلات مبنياً على هذه القيم التي تغيرت. وهناك عيب آخر مثل الموجود مع المناهج الهرمية والشبكية حيث تصميم قاعدة البيانات بأي الملفات يجب أن يسبق وصلها والنتائج التي تخزن في ملفات الوصل.

ولكن كما سبق أن ذكرنا في بداية هذا الفصل لا يوجد تعريف ثابت للمنهج شبه العلاقي. مثال ذلك يمكن أن تسمح بعض هذه النظم بانتاج ملفات وصل ديناميكياً في نفس وقت حدوث الاستفسارات.ويجب أن تعتمد تأثيرات الأداء لمثل هذا الإجراء بوضوح على تعقيدات النظام الخاص المشمول. وهناك صيغة أخرى وهي نظام يسمح بوجود ملفات وصل يسبق انتاجها ويكون لها امكانية وصل طائرة حقيقية. ويجب على مثل هذه النظم أن تقسم على الأرجح على أنها علاقية بدلاً من أنها شبه علاقية.ولكن عند هذه النقطة تصبح العناوين التي نعطيها للنظم غير مهمة فليس هناك ضرر بالنسبة للحدود الموجودة بين المناهج المختلفة. وبالرغم من ذلك فهناك نظم قواعد بيانات كافية موجهة للملف الخطي ولديها نوع ملف الوصل فقط لإمكانية الوصل والتي نشعر بأنها تستحق لاعتبارها في فئة خاصة بها تحت عنوان شبه العلاقية.

قاعدة بيانات النظام 38 لشركة IBM

مقدمة :

نظام 38 لشركة IBM (IBM System/ 38) هو نظام كمبيوتر متوسط الحجم والذي يعتبر حديثاً من الناحية المعمارية في عدد من الطرق. واحد مثل هذه الأوجه لنظام 38 الذي يهمنا هنا: أنه يصمم، كجزء متكامل من فلسفة تعميمه، لإنتاج

بيئة قاعدة بيانات وصديقة للمستفيد فعلاً وذلك لكل المستفيدين منه. وكما سوف نوضح فإن منهج قاعدة البيانات الذي ينتهجه نظام 38 عبارة عن صيغة من صيغ المنهج شبه العلاقي.

وبالحركة السريعة عبر المنطقة المعتادة فعلاً دعنا نبدأ ببعض المفاهيم الأساسية. تخزن البيانات في نظام 38 فيما سبق أن أشرنا إليه بملفات خطية بسيطة. وبالحفاظ على المصطلحات النمطية لقواعد البيانات فتسمى هذه الملفات بالملفات الطبيعية أو الواقعية physical files. ويمكن إعداد فهارس طبقاً لأي حقل أو أي خليط من حقول الملفات الطبيعية. (وتعد الفهارس في صورة تكوينات تسمى أشجار ثنائية binary trees والتي سوف نوجه لها المناقشة حتى نهاية هذا الفصل). تضاف السجلات الجديدة الى الملف عند نهايته ويضيف النظام بصورة تلقائية سجلات الفهارس المناظرة لكل الفهارس التي تشير إلى هذا الملف وبالطبع يضع سجلات الفهارس الجديدة هذه في مواقعها الترتيبية المناسبة في الفهارس. وعلى هذا فأثناء عدم وجود الملف نفسه مرتباً طبقاً لأي فئة من قيم أحد الحقول فإن الفهارس تقدم هذا الترتيب بكفاءة.

يمكن استرجاع السجلات من الملفات في نفس التسلسل الواقعي المخزنة به مباشرة (ويقول العاملون في نظام 38 عشوائياً هنا) باستخدام فهرس مناسب معداً على الملف وفي تسلسل قيمة حقل عبر أي من الفهارس المعدة على الملف أو مباشرة بواسطة الرقم النسبي للسجل. لاحظ أن استرجاع سجل بواسطة رقمه النسبي يشتمل على استرجاع سابق للسجل الذي لاحظناه ويحفظ رقمه النسبي في مكان ما للاستخدام المستقبلي.

الملفات الطبيعية والمنطقية:

مثلما يتبع اصطلاح نظام 38 «الملف الطبيعي physical file» الاستخدام النمطي لقاعدة البيانات تماماً فإن اصطلاح نظام 38 «الملف المنطقي logical file» يفعل نفس الشيء. الملف المنطقي هو نظرة المبرمج للبيانات وهو عبارة عن تخطيط لأحد الملفات الطبيعية أو لأكثر من ملف طبيعي.

الملف المنطقي المبني على ملف طبيعي واحد: يمكن أن يقدم الملف المنطقي المبني على ملف طبيعي واحد نظره للمبرمج عن البيانات التي تختلف بصورة

	PLAYNUM	PLAYNAME	AGE
1	358	Stevens	21
2	523	Doe	32
3	1131	Johnson	28
4	1779	Jones	25
5	2007	Dobbs	27
6	4280	Cohen	25
7	4319	Ross	24
8	5410	Smith	27
9	6564	Linton	24
10	8093	Smith	21
11	8366	Gomez	33

شكل 9.3
الملف الطبيعي للاعبين

معنوية عن الملف الطبيعي الأصلي. ويمكن أن يحتوي على فئة جزئية فقط من حقول الملف الطبيعي أو فئة جزئية من السجلات أو كلا من الفئتين الجزئيتين. ويمكن أن يقدم للملف ترتيباً للحقول التي يعاد ترتيبها. ويمكن أن يقدم سجلات الملف في تسلسل مختلف طبقاً لفهرس موجود على الحقل الذي يقدم هذا التسلسل.

ويبين شكل 9.3 ملف اللاعبين players الأصلي والمأخوذ من شكل 8.5. دعنا نفترض أن هذا الملف هو الملف الطبيعي لنظام 38 اللاعبين. يبين شكل 9.4a ملفاً منطقياً مبنياً على هذا الملف الطبيعي تظهر فيه السجلات في ترتيب تصاعدي طبقاً للعمر وداخل العمر طبقاً للاسم الأخير للاعب. ويشمل ذلك أنه هناك شجرة فهرس ثنائية مبنية على تسلسل حقلي Age و playname. ويبين شكل 9.4b ملفاً منطقياً ترك حقل AGE خارجاً منه مع عكس ترتيب PLAYNUM و PLAYNAME. ويبين شكل 9.4c ملفاً منطقياً يظهر فيه اللاعبون الذين تكون أعمارهم 27 سنة أو أقل. بالإضافة إلى ذلك لا يظهر حقل PLAYNAME كما أن الحقلين الآخرين موجودين في ترتيب عكسي.

أوصاف الملفات الطبيعة والمنطقية تحدد في مواصفات وصف البيانات Data Description Specifications (DDS). وبالإضافة إلى المواصفات المعتادة لخواص الحقول يمكن لمواصفات وصف البيانات DDS أن تحتوي على عناوين أعمدة ومعلومات نصية أخرى وحدود مدى للحقول العددية ومواصفات للتأكد من الصحة. وعندما يريد المبرمج كتابة شفرة للاتصال بحقل أو ملف من برنامجه

PLAYNUM	PLAYNAME	AGE
8093	Smith	21
358	Stevens	21
6564	Linton	24
4319	Ross	24
4280	Cohen	25
1779	Jones	25
2007	Dobbs	27
5410	Smith	27
1131	Johnson	28
523	Doe	32
8366	Gomez	33

(a)

PLAYNAME	PLAYNUM
Stevens	358
Doe	523
Johnson	1131
Jones	1779
Dobbs	2007
Cohen	4280
Ross	4319
Smith	5410
Linton	6564
Smith	8093
Gomez	8366

(b)

AGE	PLAYNUM
21	358
25	1779
27	2007
25	4280
24	4319
27	5410
24	6564
21	[illegible]

(c)

شكل 9.4: ثلاثة ملفات منطقية مبنية على شكل 9.3. (أ) الملف المنطقي للاعبين مخزناً طبقاً للعمر واسم اللاعب (ب) الملف المنطقي للاعبين مع عدم وجود حقل العمر مع إعادة ترتيب الحقول الأخرى (جـ) الملف المنطقي للاعبين يحتوي على السجلات التي يكون حقل العمر فيها 27 أو أقل فقط مع إعادة ترتيب الحقول الأخرى بعد إهمال أحد الحقول.

فيمكنه أن يشير فقط إلى مدخل مواصفات وصف البيانات DDS الخاصة بالملف المناسب (سواء كان ملفاً منطقياً أو ملفاً طبيعياً). وذلك بدلاً من كتابة وصف هذا الحقل أو هذا الملف من البداية. وعند ترجمة البرنامج فإن النظام ينسخ المعلومات وثيقة الصلة بالموضوع عن الحقول والملفات من DDS داخل البرنامج ويترجمها مع بقية البرنامج. وتحفظ البيانات الأخرى في مواصفات وصف البيانات DDS مثل التأكد من المدى ومعلومات التأكد من الصحة، مع الملفات.

سمة أخرى لنظام 38 هي ملف حقل الإشارة Filed Reference File (FRF) وهي صيغة لمجموعة التحكم. وهذه السمة تحفظ معلومات عن نطاق قيم الحقول في ملفات النظام 38. وإحدى سمات البيانات الهامة لنظام 38 هي أنه خلال FRF يمكن التحكم في الحقول المختلفة التي لها أسماء مختلفة بنفس تعريف النطاق..

	TEAMNUM	TEAMNAME	TEAMCITY	TEAMMGR
1	12	Dodgers	Los Angeles	Wilson
2	15	Giants	San Francisco	Johnson
3	20	Yankees	New York	Simpson
4	24	Tigers	Detroit	Corbin

(a)

	TEAMNUM	COACHNAM	COACHPH
1	12	Adams	555-1524
2	12	Baker	555-3690
3	15	Taylor	555-4820
4	20	Jackson	555-2444
5	24	Victor	555-9327
6	24	Dooley	555-7321

(b)

شكل 9.5

(أ) الملف الطبيعي للفرق
(ب) الملف الطبيعي للمدربين

وتؤخذ معلومات النطاق من FRF وتستخدم في مراقبة صحة البيانات التي ندخل في الملفات وتعرض من الملفات. والنتيجة هي مستوى مرتفع من تكامل (سلامة) البيانات.

الملف المنطقي المبني على أكثر من ملف طبيعي واحد (الوصل): تظهر صورة محددة من الوصل عندما ينتج الملف المنطقي على أكثر من ملف طبيعي واحد. ومن المعتاد أن عملية الوصل تدمج حقول سجلات ملفين مختلفين في سجلات ملف ناتج مبني على قيم شبيهة في حصول الوصل. في النظام 38 فإن الملف المنطقي المبني على ملفين طبيعيين يترك السجلات متداخلة منطقياً مع بعضها بالنسبة لهذين الملفين الطبيعيين مع ترتيب مبني على القيم المؤثرة لحقول الوصل. دعنا نفحص أحد الأمثلة.

يبين شكل 9.5 ملفات الفرق Teams والمدربين Coaches من شكل 8.5. حقول الوصل، حقول هذين الملفين التي تربط سجلات أحدهما بالآخر، من الواضح أنها حقلي TEAMNUM وحقل TEAMNAM في جدول المدربين يشمل أي الفرق التي يعمل بها مدرب محدد. ويبين شكل 9.6 ملف منطقي لنظام 38 مبنياً على الملفين الطبيعيين للفرق Teams والمدربين Coaches والسجلات مرتبة (مع ترحيل سجلات المدربين للتوضيح في هذا الشكل) بحيث أن سجلات المدربين لفريق محدد تتبع سجل هذا الفريق. وكل سجلات الفرق يمكن

12	Dodgers	Los Angeles	Wilson
12	Adams	555-1524	
12	Baker	555-3690	
15	Giants	San Francisco	Johnson
15	Taylor	555-4820	
20	Yankees	New York	Simpson
20	Jackson	555-2444	
24	Tigers	Detroit	Corbin
24	Dooley	555-7321	
24	Victor	555-9327	

شكل 9.6
ملف منطقي مبني على الملفات الطبيعية للفرق وللمدربين

استرجاعها بالترتيب وكل سجلات المدربين يمكن استرجاعها بالترتيب كما أن سجلات الفرق والمدربين بالرغم من أنها تمثل نوعي سجلات مختلفين إلا أنه يمكن استرجاعها بالترتيب المبين في شكل 9.6 على أن تتبع سجلات المدربين الذين يعملون في فريق معين سجل هذا الفريق.

استرجاع البيانات ومعالجتها:

هناك طرق عديدة لاسترجاع البيانات ومعالجتها في بيئة النظام 38. إحدى هذه الطرق تكون مع ما سبق الإشارة له في هذا الكتاب بأنه أوامر محتواة. وفي الواقع يمكن الاتصال بالملفات المنطقية والملفات الطبيعية لنظام 38 وتعديلها بعمليات استرجاع وإدخال وحذف بيانات بلغة RPG III أو لغة الكوبل أو لغة PL/I أو لغة البيسك. مرة أخرى نلاحظ أن وصف الملفات يمكن أن يؤخذ مباشرة من DDS عند ترجمة البرنامج.

هناك وسيلة اتصال بالبيانات أخرى تسمى منفعة ملف البيانات Data File Utility (DFU) والتي كما يظهر من اسمها تكون عبارة عن برنامج ذو أغراض خاصة للفحص في البيانات وتعديلها في ملفات نظام 38. كما أن جعل التطبيقات سهلة Applications Made Easy (AME) هو منتج تطبيقات قادر على إنتاج شفرة مرتبة بلغة RPG III من مواصفات التطبيقات. وعلى هذا فهو وسيلة كذلك للعمل مع ملفات نظام 38.

أخيراً فإن بيئة نظام 38 تشمل تسهيلات استفسار Query Facility صممت للاستفسارات الخاصة في وسط تشغيل الوقت الحقيقي. وهي وسيلة تدار بواسطة القوائم وتقود المستفيد من خلال سلسلة من أشكال تعرض على الشاشة مثل انتاج اجماليات لفئة حقول عددية إلى ما هو مطلوب. وتشمل تسهيلات الاستفسار هذه إمكانية الاتصال بسجلات في ملف واحد طبقاً لبيانات تسترجع في ملف آخر وهذه إحدى صور الوصل.

شجرة الفهرس الثنائية.

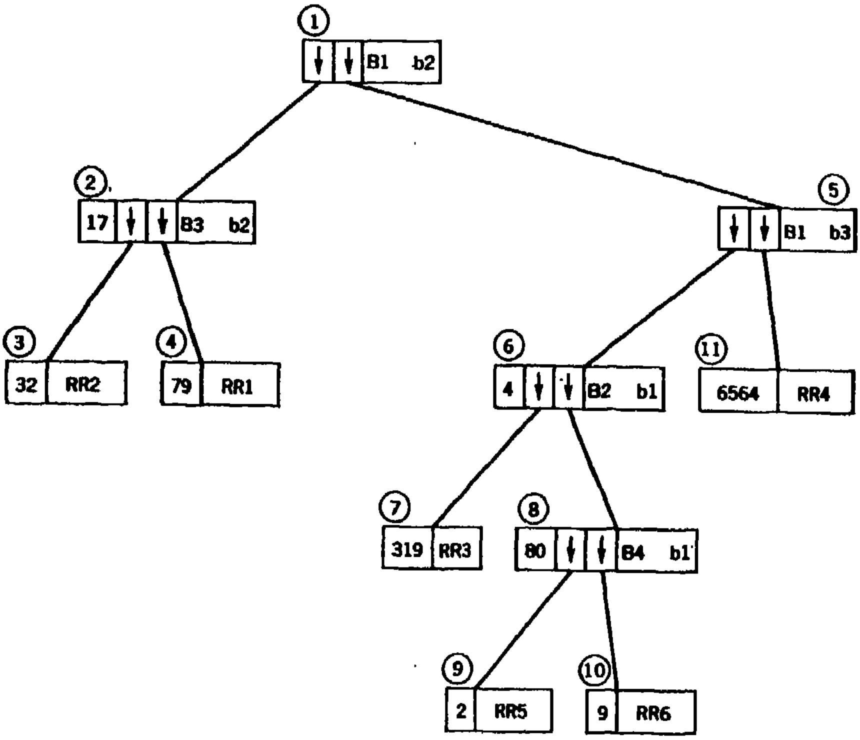
مقدمة: تعد الفهارس في نظام 38 على هيئة أشجار (أو بدقة أكثر أشجار ثنائية كاملة). والشجرة الثنائية (الكامل). هي شجرة تكون كل عقدة فيها لها فرعين منبثقين لأسفل منها أو أنها تكون عقدة بداية أو نهاية لا توجد فروع منبثقة منها لأسفل. وكما أن شجرة الفهرس الثنائية لنظام 38 عبارة عن فكرة جديدة ومختلفة جداً عن تكوينات الفهارس التي ذكرت في الفصل الثالث فإننا نستطرد ونصفها في محتوى أحد الأمثلة.

شكل 9.7 هو جزء من ملف اللاعبين Players لمثال البيسبول. ولغرض التوضيح فإنه يشمل بعض السجلات من جدول اللاعبين Players الموجود في شكل 8.5 والبعض الآخر كذلك. لاحظ أيضاً أنه ليس متسلسلاً طبقاً لحقل PLAYNUM (أو أي حقل آخر) والذي يجعل النقطة أن الملف في نظام 38 لا يحتاج لأن يكون في أي تسلسل طبيعي.

شكل 9.8 هو شجرة فهرس ثنائية مبنية على حقل PLAYNUM لملف اللاعبين Players الموجود في شكل 9.7. سوف نصف أولاً الشكل الغريب للبيانات في هذه الشجرة ثم نناقش بعد ذلك كيفية أداء بحث خلالها وأخيراً توضح كيفية أعدادها.

	PLAYNUM	PLAYNAME	AGE
1	1779	Jones	25
2	1732	Blake	22
3	4319	Ross	24
4	6564	Linton	24
5	4802	Turner	33
6	4809	Norton	28

شكل 9.7
جزء من ملف اللاعبين



شكل 9.8
شجرة فهرس ثنائية معدة على حقل Playnum لملف اللاعبين الموجود في شكل 9.7

تكوين الشجرة: أولاً الأرقام الموجودة داخل دوائر والتي تسمى العقد في شكل 9.8 هي موجودة ببساطة لتريحنا في هذا التوضيح. وحيث أن التكوين عبارة عن شجرة ثنائية كاملة فيجب أن يوجد سهمين مثل هذه الأسهم بالضبط في كل عقدة ليست عقدة بداية أو نهاية. والأرقام الموجودة على يسار بعض العقد هي قيم كلية أو جزئية لحقل (أو حقول) البيانات (ويشار إليها كذلك بأنها حقل رئيسي Key filed) وتكون مفهرسة، في حالتنا هذا الحقل PLAYNUM الموجود في ملف اللاعبين Players. والقيمة الجزئية التي يمكن أن تظهر في عقدة غير عقدة البداية أو نقطة النهاية فقط تعرف بأنها بيانات مشتركة Common data. وكما سنوضح فهي تكون جزءاً من قيمة كل المفاتيح الممثلة أسفلها في الشجرة. وتشير قيم RR إلى الأرقام النسبية للسجلات الخاصة بملف اللاعبين. مثال ذلك لاحظ أن العقدة 11 تحدد أن السجل الذي قيمه PLAYNUM هي 6564، يوجد في موضع له رقم سجل نسبي 4 في ملف Players إرجع مرة أخرى إلى شكل 9.7 للتأكد من

ذلك. تذكر أن هذا يعني أن سجل Linton كان السجل الرابع الذي أضيف إلى هذا الملف وهو السجل الرابع واقعياً في الملف الخطي. بعض قيم الحقل تم إعدادها بصورة تقدمية بدءاً من الجذر الخامس بشجرة الفهرس. مثال ذلك الحركة من العقدة 1 إلى العقدة 2 إلى العقدة 3 تجعلك تعد قيمة الحقل 1732. وفي عقدة النهاية تجد أن السجل الذي له PLAYNUM مساوياً لهذه القيمة هو السجل الثاني في ملف اللاعبين. ويشير RR إلى ملف واقعي يمكن أن يظهر في عقد النهاية فقط. وتمثل حروف b الكبيرة والصغيرة بت داخل بايت خاصة بقيم الحقل الذي يختبر عند توجيه حاجز خلال الشجرة (وهو الرمز الذي نستخدمه ولا يوجد في المراجع الخاصة بنظام 38).

وتمثل B بايت يعد من اليسار بدءاً بالرقم 1. أما b فتمثل بت تعد من اليسار وتبدأ للتبسيط بالرقم 1 الذي يمثل البت الموجودة على أقصى يسار الأربعة بت التي تمثل الرقم من 0 إلى 9 تمثيلاً ثنائياً. بالطبع هذه تمثل الأربعة بت الموجودة على اليمين فقط في البايت أما الأربعة بت الموجودة في يسار البت فهي متطابقة لكل الأرقام ولا تؤثر على ذلك على النتيجة كما سيتضح ذلك لنا.

البحث في الشجرة: تذكر أن الغرض من الفهرس هو السماح بالاتصال المباشر بسجل موجود في ملف وذلك عبر قيمة إحدى حقوله. وللسماح باسترجاع سجلات الملف تتابعياً طبقاً لقيمة إحدى حقوله دعنا نوضح اتصالاً مباشراً في شجرة فهرس ثنائي باستخدام مثال. إفرض أننا نبحث عن سجل في ملف اللاعبين له القيمة 4802 للحقل PLAYNUM. نقوم بإدخال الفهرس عند عقدة الجذر (العقدة 1) . هذه العقدة تخطرنا بأن نختبر البت 2 من البايت 1. (سوف نناقش إعداد هذا الاختبار في القسم التالي). في مثل هذا الاختبار للبت تكون القاعدة العامة هي أن 0 يرسلنا إلى اليسار بينما 1 يرسلنا إلى اليمين وحيث أن بايت 1 من 4802 هي 4 وبت 2 من 4 (0100 = .4) هي 1 فإننا نتحرك لأسفل إلى اليمين. وهناك نلتقي بالعقدة 5. الاختبار على البايت 1 (نفس 1 لـ 4) وبت 3 يؤدي إلى 0 ويحركنا لأسفل إلى اليسار. وهذا يحضرنا إلى العقدة 6 حيث نجد لأول مرة بيانات مشتركة وهي 04. هذا يعني أن كل العقد من هذه العقدة

ولأسفل في الشجرة قيم PLAYNUM التي لها الرقم 4 كأول رقم من على أقصى اليسار. لماذا الموقع الموجود على أقصى اليسار؟ لأن 4 هذه هي أول بيانات مشتركة حصلنا عليها عند التتبع من الجذر ولأسفل. فإذا لم تناظر البيانات المشتركة البيانات الموجودة في نفس موقع البايت في قيمة الحقل الذي نبحث عنه فلا يكون مسموحاً لنا بعمل اختبار بت محدد ونقرر أن شرط عدم الحدوث لقيمة هذا الحقل قد حدث وننهي البحث في الفهرس. لكن في واقع الأمر أول بايت من 4802 هو 4 وعلى هذا فإننا نختبر بايت 2 (وهي 8) بأخذ أول بت منها واختباره ونجد أن أول بت هو 1 والذي ينقلنا لأسفل وإلى اليمين. وعند العقدة 8 نجد 80 على أنها بيانات مشتركة ونلاحظ أنها تناظر 80 في الموقع الثاني والثالث للبت لقيمة حقل البحث 4802. اختبار بايت 4 (وهو 2) بأخذ أول بت منه ينتج عنه 0 والذي يرسلنا لأسفل وإلى اليسار إلى عقدة النهاية 9. والرقم 2 الذي نجده هناك يكمل القيمة 4802. وحيث أن العقدة 9 هي نقطة نهاية فإذا لم يكن هناك 2 فيكون شرط عدم الوجود قد تحقق لكن عند العقدة 9 يذكر أن السجل الذي له PLAYNUM يساوي 4802 يكون له موقع سجل نسبي 5 في ملف اللاعبين. وقد تبدو هذه العملية معقدة لما يتوقع أن يكون عملية بحث مفهرس مباشر وسريع لكن تذكر أنه عند كل عقدة غير عقد النهايات يكون الاختبار الذي يؤدي على بت واحدة فقط والتعليمات المخزنة طبقاً لهذه البت توضح هناك عند وقت إعداد شجرة الفهرس الثنائي.

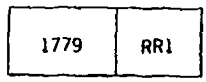
لاستخدام شجرة الفهرس الثنائي في استرجاع كل السجلات المرتبة فإننا نعكس من القمة إلى القاعدة ومن اليسار إلى اليمين وهذا هو نفس ترتيب ترقيم العقد. إقنع نفسك بأن مثل هذا العكس سوف يؤدي لإيجاد قيم حقل PLAYNUM (والسجلات المناظرة له في ملف اللاعبين من خلال الأرقام النسبية للسجلات) في ترتيب تصاعدي.

إعداد الشجرة: أخيراً هناك سؤال خاص بكيفية إعداد شجرة الفهرس الثنائي في بداية إعدادها. إعداد شجرة فهرس في نظام 38 هو في واقع الأمر عملية تنفيذ لإدخال متعدد لقيم حقل رئيسي داخل عقد جديدة. وفي الأساس لأداء أي إدخال (إضافة) يبدأ النظام عند الجذر ويبحث خلال الشجرة مع

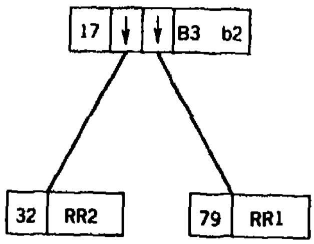
افترض أن العقدة التي لها هذه القيمة موجودة بالفعل في الشجرة وهناك محاولة لاسترجاعها. وعند الوصول إلى عقدة نهاية فتستبدل عقدة النهاية هذه بعقدة غير عقد النهايات والتي يكون لها قيمة النهاية القديمة والقيمة الجديدة المراد إدخالها عند عقدتي نهاية الانطلاق لأسفل من عند هذه العقدة وكما سوف نرى فهناك أكثر من ذلك قليلاً.

دعنا نفترض أننا سنبدأ في بناء ملف اللاعبين Players الموجود في شكل 9.7 وإن شجرة الفهرس الثنائي تعد على الحقل PLAYNUM من البداية. أولاً سجل اللاعب Jones يضاف إلى ملف اللاعبين (الجديد) وتشبه شجرة الفهرس الثنائي التكوين البسيط الموجود في شكل 9.9a . وهي عقدة فردية تحدد أن السجل الذي له PLAYNUM قيمته 1779 (السجل الخاص باللاعب Jones) يكون في موقع السجل النسبي 1 في ملف اللاعبين. يلي ذلك إضافة سجل Blake إلى الملف وقيمة حقل PLAYNUM لهذا السجل هي 1732 ويقارن مع سجل الفهرس الواحد الموجود في شكل 9.9b أنه تم انتاج جذر جديد له 17 كبيانات مشتركة. أول بايت من ناحية اليسار تختلف بين 1732 و 1778 هي 3 بمقارنة مكونات 3 من البت (0011) نجد أول بت مختلف من ناحية اليسار هو البت 2. وعلى هذا فالقيمة التي لها بت 0 ترسل الى اليسار والقيمة التي لها بت 1 ترسل إلى اليمين لهذا يكون المفتاح الجزئي 32 والذي يكمل القيمة 172 بالسجل النسبي 2 يرسل لأسفل ولليسار أما 79 الخاصة بالرقم 1779 فترسل لأسفل لليمين في شكل 9.9b معلومات اختبار B3b2 والبيانات المشتركة تخزن في عقدة غير نهائية تفصلهما. يلي ذلك سجل Ross والذي له القيمة 4319 للحقل PLAYNUM ويضاف إلى ملف اللاعبين. لإيجاد الموقع المناسب لإضافته في الشجرة في شكل 9.9b سنفعل ما سبق إن فعلناه بطريقة بسيطة بعض الشيء وذلك عند إعداد شكل 9.9b. أي إننا نتتبع خلال الشجرة كما لو كنا نبحث عن القيمة التي نريد إضافتها وعند ذلك نضيفها بالفعل عندما نصل إلى عقدة نهاية كما سبق ذكر ذلك (العقدة الوحيدة التي كانت في شكل 9.9a هي عقدة نهاية). وعلى هذا فإننا نحاول أن نفعل نفس الشيء مع القيمة 4319 والشجرة الموجودة في شكل 9.9a لكننا لا نستطيع أن ننتقل لأسفل من عقدة الجذر نظراً لأن البيانات المشتركة المحددة فيها (17) لا تناظر أول مواقع لاثنين بايت للقيمة 4319 التي نحاول أن

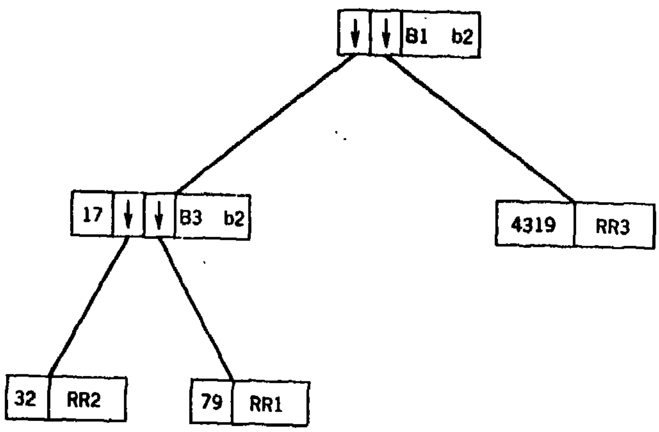
نضيفها إلى الشجرة. عندما يحدث ذلك في إحدى محاولات الإدخال فتبنى الشجرة لأعلى من عند هذه العقدة. وعلى هذا تبين الشجرة الموجودة في شكل 9.9c جذراً جديداً يميز بين 1 للبيانات المشتركة 17 للجذر القديم و 4 للقيمة الجديدة 4319. وهكذا خلال بقية السجلات التي تضاف.



شكل 9.9a. تطوير شجرة فهرس ثنائية : أول سجل



شكل 9.9b. : تطوير شجرة فهرس ثنائية : بعد إضافة السجل الثاني



شكل 9.9

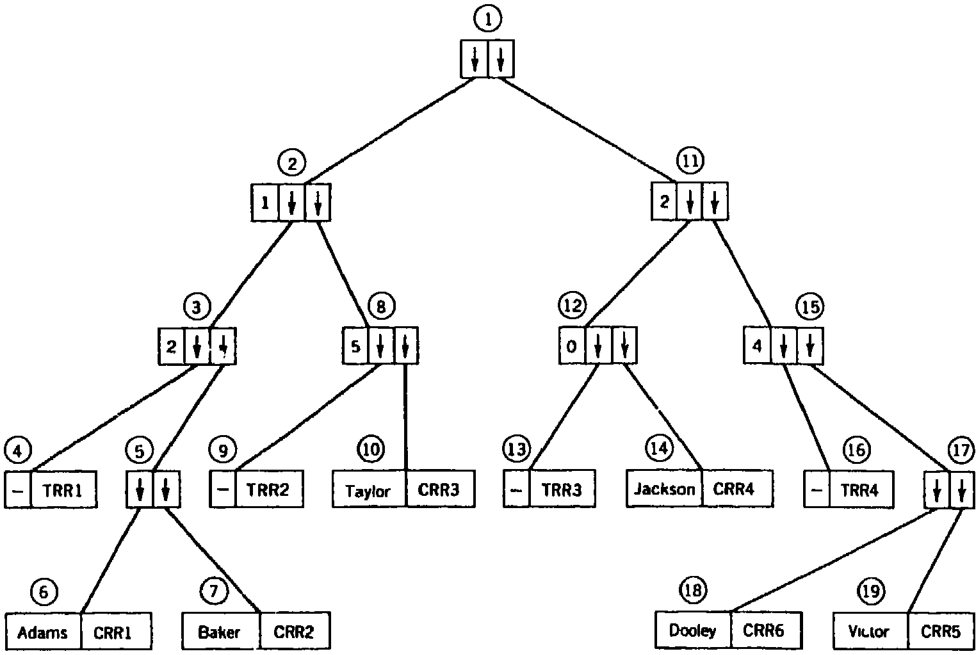
تطوير شجرة فهرس ثنائية: بعد إضافة السجل الثالث

وهناك أكثر من ذلك إلا أن هذه هي الفكرة الأساسية. ونعود إلى أن الطريقة التي تم وصفها يمكن أن تضع بعض القيم خارج ترتيب متتابع من القمة

إلى القاعدة ومن اليسار إلى اليمين. وعندما يحدث ذلك يستدعي النظام 38 تعليمات أخرى تدخل تغييرات في الترتيب بحيث أنه يتم الحفاظ على الترتيب المناسب. كما يمكن أن تسأل عن كيفية أو مدى كفاءة عملية بحث الفهرس إذا كانت محتويات الشجرة ليست في الذاكرة الرئيسية أثناء البحث وهذا شيء جيد بالنسبة للملف الكبير. والإجابة تقع في الطريقة الذكية لتجميع أجزاء من الشجرة في الذاكرة الرئيسية عند الحاجة لذلك.

تنفيذ وصل باستخدام الشجرة: يبين شكل 9.10 منظراً لشجرة بحث ثنائي تجعل الملف المنطقي الموجود في شكل 9.6 ممكناً. مرة أخرى أسماء العقد الموجودة داخل دوائر هي لأغراض الإشارة فقط. وقد تركنا معلومات اختبار البت عند عقد غير نهائية وذلك للتبسيط. وفي العقد النهائية فإن «T» و «C» المساعدات إلى أرقام RR تشير إلى ملفات Teams و Coaches على التوالي والحقول المشمولة في هذه الشجرة هي حقول TEAMNUM لكل من الملفات الطبيعية Team و Coaches وحقل COACHNAM للملف Couches. دعنا نتتبع خلال جزء من هذه الشجرة. تبين العقدة 2 بيانات مشتركة من 1 تحدد أن كل العقد الموجودة تحتها تقود الى سجلات teams و Coaches تشمل فرقاً يبدأ رقمها بالقيمة 1. العقدة 3 تبين بيانات مشتركة من 2 في الموقع الثاني لبايت رقم الفريق. بدمج معلومات العقدة 2 والعقدة 3 فإننا نتحقق أن كل العقد الموجودة أسفل العقدة 3 لديها شيء معين مع الفريق 12 وهو Dodgers. العقدة 4 هي عقدة نهاية وتبين فراغ (الشرطة الموجودة هناك) حيث يوجد اسم كامل أو اسم جزئي لمدرب. وهذا يعني أن العقدة 4 تشير إلى سجل من سجلات الفريق 12 نفسه وتؤكد أن العقدة لها مشير إلى موقع نسبي 1 لسجل من ملف Teams (TRR1) حيث يوجد السجل الخاص بالفريق 12. والحركة من عقدة النهاية 4 من أعلى لأسفل ومن اليسار لليمين بالنسبة للعقد العاكسة فإننا نجد أن العقدة 5 تقود إلى العقدتين 6 و 7 والتي تمثلان في الواقع وتحتويان على مشيرات الى سجلات مدربين لكل من Adams و Baker وهما مدربي Dodgers. العقدة التالية في الترتيب هي العقدة 8 والتي تقود الى عقد محددة لرقم الفريق 15 ومدربيه وهكذا.

ملاحظة: عند وقت الطباعة أعلنت شركة IBM عن وسيلة جديدة أكثر عمومية لنظام 38 تضاف إلى التسهيلات الموجودة فعلاً.



شكل 9.10

شجرة فهرس ثنائية للملف المنطقي الموجود في شكل 9.6

مراجع: References

IBM Corporation, IBM System/38 Control Program Facility Concepts Mannual, IBM Form No. GC21 - 7729, 1981.

IBM Corporation, IBM System/38 Control Program Facility Programmer's Guide, IBM Form No. SC21 - 7730, 1981.

IBM Corporation, IBM System/38 Introduction, IBM Form No. GC21 - 7728, 1983.

أسئلة وتمارين: Questions and Exercises

١ ـ وضح العبارة التالية: «المنهج شبه العلاقي لقاعدة البيانات هو هجين بين المنهج العلاقي والمنهجين الهرمي والشبكي».

٢ ـ ما ملف الوصل في المنهج شبه العلاقي؟ ما المميزات والعيوب المصاحبة له؟

٣ ـ كيف يميز النظام 38 بين الملفات المخزنة ووجهة نظر المبرمج لهذه الملفات؟ وما هو معنى هذا التمييز؟

٤ ـ إرسم منظراً منطقياً للملف في النظام 38 الطبيعي الموجود في شكل 9.3 حيث يأتي اسم اللاعب في الملف أولاً يليه عمره مع إهمال رقم اللاعب والسجلات مرتبة ترتيباً أبجدياً طبقاً لاسم اللاعب ولا يوجد سوى اللاعبين الذين تزيد أعمارهم عن 27 سنة فقط في هذا الملف.

٥ ـ صف الصيغة المحددة للوصل في قاعدة بيانات النظم 38.

٦ ـ بكلمات من عندك صف كيف تعد شجرة الفهرس الثنائي في نظام 38.

٧ ـ ما مميزات منهج شجرة الفهرس الثنائي ؟

٨ ـ إرسم شجرة فهرس ثنائي لنظام 38 لها القيم الرئيسية التالية بالترتيب المبين:

a. 55237
b. 8326
c. 5528
d. 1274
e. 1500
f. 5524
g. 7731

٩ ـ كيف تستخدم شجرة الفهرس الثنائي لنظام 38 في تحقيق وصلاً في قاعدة بيانات النظم 38؟

١٠ ـ صف كيف يمكن أن يجد النظام 38 سجل Doley في شجرة البحث الثنائي الموجودة في شكل 9.10.

الفصل العاشر
تصميم قاعدة البيانات

مفهوم تصميم قاعدة البيانات:

The Concept of Database Design

مقدمة:

يشير تصميم قاعدة البيانات الى عملية تنظيم حقول البيانات اللازمة لتطبيق وإصدار أكثر في هيكل منظم. ويجب أن يعكس الهيكل العلاقات اللازمة بين الحقول أثناء مطابقتها مع القيود الواقعية لنظام إدارة قاعدة البيانات المستخدم. وهناك جزءان للعملية عادة ما يصاحبان اصطلاح تصميم قاعدة البيانات لدى الناس. أحدهما هو تصميم قاعدة البيانات المنطقية والتي لها مكونين. الأول يشمل تنظيم حقول البيانات في مجموعات غير متكررة مبنية على علاقات البيانات. والثاني يشمل تنظيم أولى لهذه المجموعات المنطقية في تكوينات مبنية على طبيعة نظام إدارة قاعدة البيانات والتطبيقات التي سوف تستخدم البيانات. وهناك أساليب عديدة لعمل تصميم قواعد البيانات المنطقية وكل من هذه الأساليب له اهتمام خاص ومنهج خاص. والجزء الثاني من تصميم قاعدة البيانات هو تصميم قاعدة البيانات الطبيعية والذي يشير إلى إعادة ضبط التكوينات التي سبق وصف استخلاصها وذلك لتطبيق خواص أداء وتشغيل نظام إدارة قاعدة البيانات ومرة أخرى يكون هذا موجهاً عن طريق متطلبات تشغيل التطبيق.

لكن يبدأ الأعداد لتصميم قاعدة البيانات في مرحلة سابقة من مراحل دورة حياة تشغيل البيانات. أو بوضعها بطريقة أخرى كيف نعرف أن إدخال الحقول والعلاقات في قاعدة البيانات عن طريق عملية تصميم قاعدة البيانات كانت في

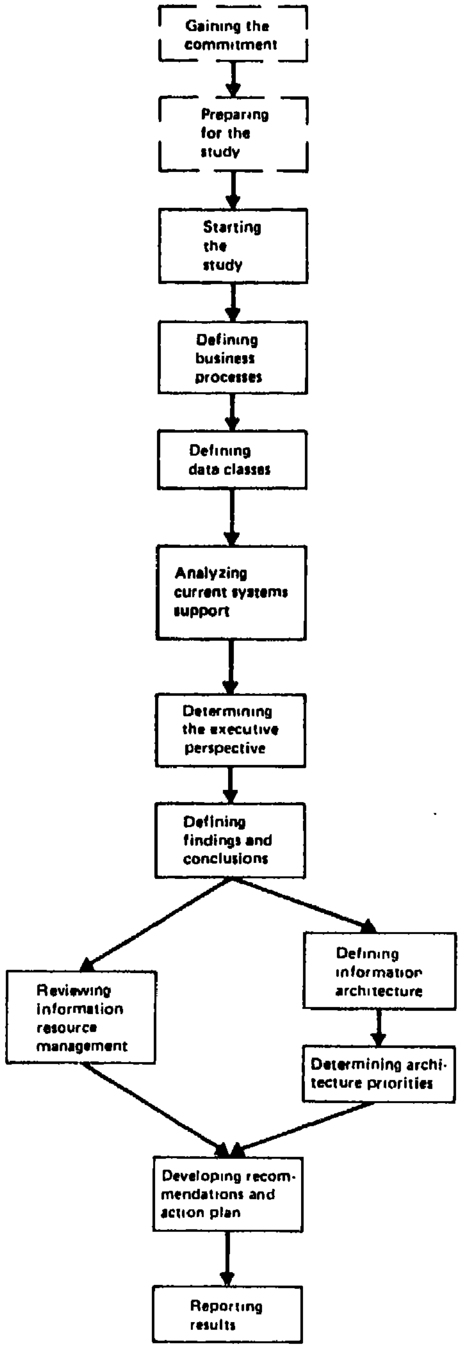
المقام الأول؟ وتبدأ القصة عند مرحلة تسمى بالتخطيط الاستراتيجي لنظم المعلومات.

التخطيط الاستراتيجي لنظم المعلومات:

لقد كان تاريخ تشغيل البيانات مثل تاريخ التطبيقات يحدث له تطوير مستقل لكل نظام من النظم الأخرى: لقد سبق أن ناقشنا بالفعل إحدى نتائج ذلك التكرار على مستوى واسع للبيانات في الملفات المبعثرة بدلاً من المشاركة في البيانات كلما كان ذلك ممكناً. ويجب أن تأتي مرحلة في كل عملية نمو لتشغيل بيانات الشركة عندها يحدث تمييز أن الطريقة غير المنظمة لتطوير التطبيقات يجب أن ينتج عنها منهجاً مخططاً بصورة جيدة. وكخطوة أولى يجب أن تعد الشركة دراسة جادة لبيئة تشغيل بياناتها الحالية واحتياجاتها المستقبلية من تشغيل البيانات وطريقة تداخل مكوناتها المختلفة بالنسبة لمرور البيانات للأمام وللخلف.

وأحد الطرق المتاحة التي صممت لتحقيق هذا الغرض هي أسلوب تخطيط نظم الأعمال Business Systems Planning (BSP) الذي أعدته شركة IBM. ويبين شكل 10.1 مسار الخطوات في دراسة تخطيط نظم الأعمال BSP. ويمكن أن يكون مدى الدراسة عبارة عن داخليات الشركة أو الفرع أو بعض الواحدات العاملة الأخرى طبقاً للحكم والدراسة التي تأخذ حوالي ستة أسابيع كفترة مثالية لإتمامها يجب أن تؤديها مجموعة تتكون من 8 إلى 12 فرداً يمثلون المناطق الوظيفية الرئيسية في الشركة, ومحللي النظم الرئيسيين وأحد منفذي الإدارة العليا كعنصر هام جداً في المجموعة. وباصطلاحات تصميم قاعدة البيانات فإن الأجزاء الرئيسية لعملية BSP هي الخطوات الموجودة في شكل 10.1 المسماة لتعريف عمليات الأعمال. وتعريف فئات البيانات وتعريف معمارية المعلومات.

ويشير تعريف عمليات الأعمال إلى دراسة تفصيلية للشركة والتي تحدد مكونات عملياتها. ويشير تعريف فئات البيانات الى دراسة تعريف الأنواع الرئيسية للبيانات (ليست حتى مستوى الحقل) اللازمة للشركة وبالتالي لعمليات أعمال الشركة. ويبين شكل 10.2 عينه لمصفوفة تربط عمليات أعمال الشركة المبينة على اليسار مع فئات بياناتها المبينة في القمة. وعند نقاط التقاطع الفردية فإن تسمية C تحدد العملية المسؤولة عن إنتاج البيانات في فئة البيانات هذه بينما تسمية



شكل 10.1

خطوات BSP (أعيدت طباعتها بإذن من Information Systems Planning Guide (GE 20 - 0527 - 3) من IBM)

PROCESS \ DATA CLASS	Planning	Financial	Product	Parts master	Bill of material	Vendor	Raw material inventory	Fin. goods inventory	Facilities	Work in process	Machine load	Open requirements	Routings	Customer	Sales territory	Order	Cost	Employee
Business planning	C	U	U						U					U			U	U
Organization analysis	U																	
Review and control	U	U																
Financial planning	C	U								U								U
Capital acquisition		C																
Research			U												U			
Forecasting	U		U											U	U			
Design and development			C	C	U									U				
Product specification maintenance			U	C	C	U												
Purchasing						C											U	
Receiving						U	U											
Inventory control							C	C		U								
Workflow layout			U						C				U					
Scheduling			U			U			U	C	U					U		
Capacity planning						U			U		C	U	U					
Material requirements			U		U	U						C				U		
Operations										U	U	U	C					
Territory management			U											C		U		
Selling			U											U	C	U		
Sales administration															U	U		
Order servicing			U											U		C		
Shipping			U					U								U		
General accounting		U				U								U			U	U
Cost planning						U										U	C	
Budget accounting	U	U								U							U	U
Personnel planning		U																C
Recruiting/development																		U
Compensation		U																U

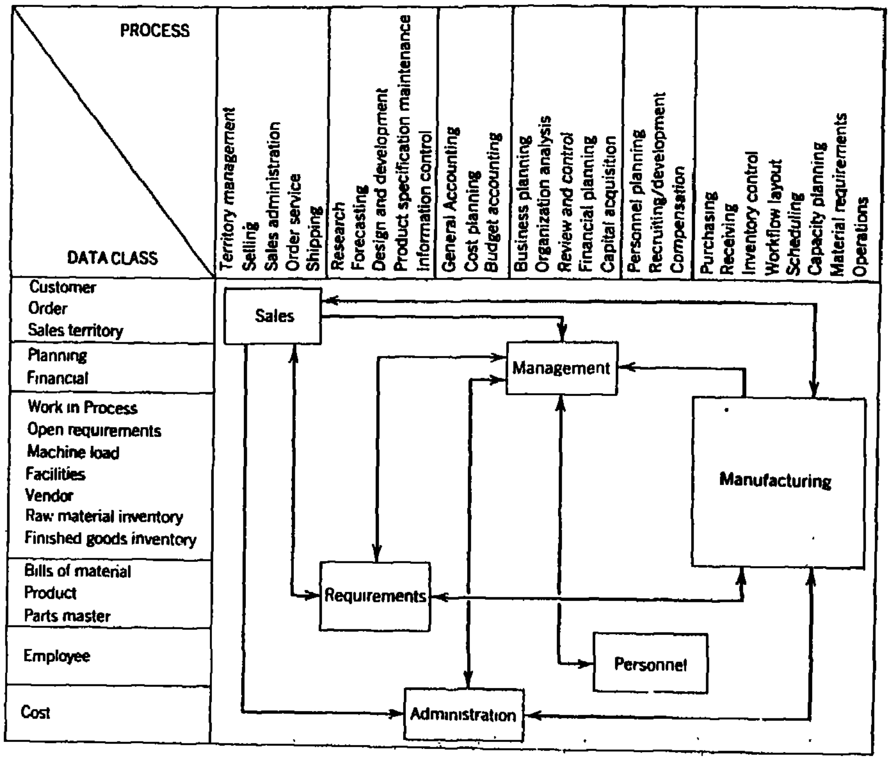
شكل 10.2

عينة لفئة بيانات مع مصفوفة عملية الأعمال من BSP (أعيدت طباعتها بإذن من Information Systems Planning Guide (GE 20 - 0527 - 3) من IBM)

U تحدد أن العملية تستخدم البيانات في فئة البيانات.. والتجميع المحدد بواسطة المستطيلات ذات الأضلاع السميكة مبني على عمليات مرتبطة مع بعضها البعض (مثال ذلك المجموعة الثالثة لأسفل مكونة من عمليات تشغيل).

ومن الواضح من مربعات U التي تقع خارج المستطيلات ذات الأضلاع السميكة في شكل 10.2 أنه في عدد كبير من اللحظات تنتمي العمليات الى إحدى المجموعات الرئيسية التي تستخدم بيانات تنتجها عمليات تنتمي الى مجموعة أخرى من المجموعات الرئيسية. مثال ذلك عملية المواد المطلوبة لمجموعة التشغيل تستخدم بيانات منتج تنتجها عملية الأبحاث والتطوير للاحتياجات (مجموعة

الأبحاث والتطوير). مثل هذا الشمول المتقاطع للبيانات عبر تجميع عمليات الأعمال طبيعي واكتشافه يكون نتيجة حاسمة لعمل BSP. شكل 10.3 (بتغيير محاور عملية الأعمال وفئة البيانات) يستمر في تطوير مصفوفة شكل 10.2 من خلال



شكل 10.3
عينة من معمارية معلومات BSP. (أعيدت طباعتها بإذن من Information Systems Planning Guide (GE 20 - 0527 - 3) من IBM)

مرحلة تعريف معمارية المعلومات. ويكون لدينا فكرة جيدة عند هذه النقطة من عمليات أعمال المنظمة وفئات بياناتها والعلاقات الموجهة ناحية البيانات داخلها.

وبتعريف المستوى الاستراتيجي هذا لمعمارية المعلومات نستطيع الآن أن نعيد انتباهنا الى مجهودات تطوير التطبيقات الفردية. وسوف نقوم بتجزئة العمليات إلى عمليات جزئية ونرتبها طبقاً لأولويات بالنسبة لتنفيذها على الكمبيوتر. وتخدم معمارية المعلومات كسجل دائم لكيفية ارتباط كل تطبيقات الكمبيوتر مع بعضها البعض في الخطة الاستراتيجية.

تحليل النظم :

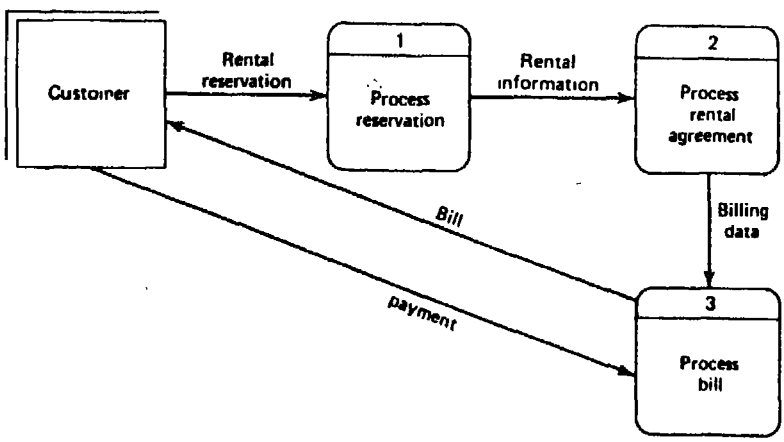
باتخاذ القرار بتنفيذ أحد العمليات الجزئية التي سبق لنا تعريفها في دراسة BSP فإننا ننتقل إلى دورة تطوير التطبيق. وتبدأ دورة تطوير التطبيق بكل أهمية مرحلة تحليل النظم. وتحليل النظم هو دراسة تفصيلية لكيفية اعتبار عملية الأعمال (وهو ما نسميه في الواقع بالعملية التفصيلية في القسم السابق) وذلك بالنسبة للتنفيذ على الكمبيوتر. وتحليل النظم هو تمرين مستقل للتنفيذ ولا يشمل ذكر الشرائط أو الأقراص أو وحدات التشغيل المركزية أو أي وحدة من وحدات نظم مكونات الكمبيوتر الأخرى. ومن الطيش فعلاً الاستمرار بتنفيذ الكمبيوتر لعملية غير مفهومة تماماً فأي اعتقاد خاطيء لكيفية عمل العملية من المؤكد ظهوره في داخل شفرة التطبيق. وكلما اكتشفت هذه الأخطاء فيما بعد في دورة تطوير التطبيق كلما كان من المكلف تصحيحها.

ويعمل محلل النظم كسطح بيني بين الأفراد المسؤولين عن أداء عملية الأعمال (المستفيدين) والأفراد المسؤولين عن تنفيذها على الكمبيوتر (المبرمجين). ويقوم محلل النظم بعمل مقابلات مع المستفيدين ودراسة وثائق عن عملية الأعمال وإعادة صياغة هذه المعلومات في صورة مناسبة وتمريرها الى المبرمجين. ويجب أن تدعم طريقة محلل النظم تنفيذ هذه الأنشطة. والطريقة وكيفية تسجيلها لمعلومات عملية الأعمال يجب أن تكون تفصيلية بدرجة كافية ليعمل بها المبرمجون. ولكن في نفس الوقت يجب ألا تكون مخيفة بدرجة كافية للمستفيدين غير المعتادين على التحليل وذلك ليكونوا قادرين على قراءة وفهم الغرض من توفير تغذية مرتجعة لدقتها. وإحدى مثل هذه الطرق تسمى بتحليل النظم المرتب Structured Systems Analysis.

وتحليل النظم المرتب موجة للبيانات وهو منهج موجه هندسياً لتحليل النظم الذي طور في نهاية السبعينيات الميلادية. ومنذ أن أصبح الاصطلاح «مرتب» من الكلمات الأكثر استخداماً في تشغيل البيانات فيتعجب الإنسان بالنسبة لاستخدام هذا الاصطلاح هنا. في واقع الأمر هذا المنهج كما سيتضح هو «مرتب» بمعنى أنه ذو طبيعة العمل من أعلى لأسفل أو من القمة إلى القاعدة كما أنه مصمم ليكون متكاملاً مع أساليب تصميم البرامج المرتبة كما أنه منهج موحد في حل مشاكل تحليل النظم.

وفي الواقع الفعلي فإن تحليل النظم المرتب عبارة عن عمل العديد من الأساليب بعضها جديد وبعضها قديم نسبياً. والمركبة الرئيسية هي رسومات مسارات البيانات Data Flow Diagram (DFD) والتي تهدف إلى التوضيح البياني لحركة البيانات خلال النظام. والوسائل الأخرى المستخدمة تشمل رسومات تكوين البيانات Data Structure Diagram وأشجار القرارات وجداول القرارات والإنجليزية المرتبة والإنجليزية المضبوطة وتطبيع البيانات وقواميس البيانات.

ورسم سريان البيانات DFD هو مكون في اتجاهين يتكون من أربعة أنواع لعناصر أساسية: صناديق عمليات (تمثل بمستطيلات رأسية لها أركان دائرية) ومخازن بيانات (تمثل بمستطيلات أفقية بها ضلع ناقص) وكينونات خارجية (مربعات) وأسهم مسارات البيانات. وتنقل صناديق العمليات البيانات بينما تقوم مخازن البيانات بحفظ البيانات وتسهل أسهم سريان البيانات حركة البيانات أما الكينونات الخارجية فهي أسطح بينية مع الأفراد أو النظم الموجودة خارج النظام الحالي. وشكل 10.4 هو مثال لرسم سريان بيانات DFD لنظام الحجز والفواتير لشركة تأجير السيارات.



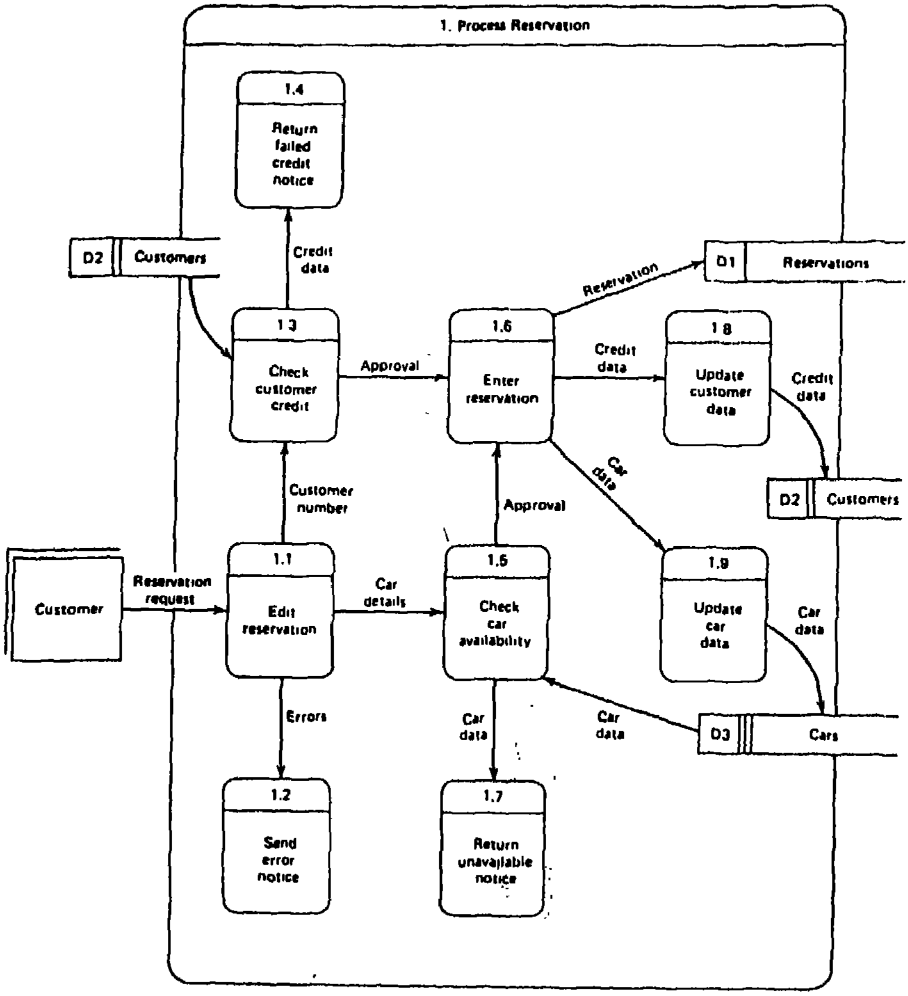
شكل 10.4
تحليل نظم مرتب: رسم سريان البيانات لنظام حجز استئجار السيارات

وتهدف رسومات مساراً لبيانات DFD إلى كونها غير طبيعية أو غير واقعية

مبينة مسار المعلومات بين العمليات فقط والتخزين والكينونات الخارجية مع عدم إظهار الشرائط أو الأقراص أو وحدات التشغيل المركزية أو أي وسيلة تنفيذ واقعية. وهي مصممة لكي يفهمها المبرمجون والمستفيدون كذلك وذلك لتسهيل عملية التغذية المرتجعة للمستفيد. ولهذا الغرض فإن أحد سمات مفهوم DFD هو أن الرسومات ترسم من القمة إلى القاعدة. وباعتراف الجميع فإن شكل 10.4 هو منظر مرتفع المستوى للنظام. إلا أنه لا يفي باحتياجين. أحد أسباب ذلك أنه يعد تخطيطاً لمحتويات النظام والسبب الآخر أنه يقدم وسيلة سهلة الفهم للمستفيدين غير الموجهين ناحية التحليل.

أما الشكل الموجود في شكل 10.5 فهو تفجير لصندوق عملية الحجز الموجود في شكل 10.4. وصناديق العمليات الموجودة في شكل 10.5 عندما تؤخذ مع بعضها تصف بتفاصيل أكثر الوظيفة المذكورة في صندوق العملية الموجودة في شكل 10.4. والمفهوم هو أن صناديق العمليات يتم تفجيرها على مستوى أقل وينتج عن ذلك هرمية من مستويات صناديق العمليات حتى يمثل صندوق عملية آخر مستوى كماً من العمل ينفذ بواسطة جزء من الشفرة (حوالي من 50 إلى 100 سطر بإحدى اللغات مرتفعة المستوى) في مفهوم التصميم المرتب. وعلى هذا بأخذ كل رسومات مسار البيانات DFD مع بعضها فإنها تصف التطبيق من القمة إلى القاعدة وصفاً شاملاً وبطريقة سهلة الفهم.

وتستخدم رسومات سريان البيانات DFD بواسطة مصممي البرامج والمبرمجين في تنفيذ التطبيق. وعلى هذا فهي تتبع أنه يجب أن يوجد مستوى آخر من المنطق التفصيلي لوصف تفاصيل الوظائف الموجودة في صناديق عمليات المستوى الأدنى. ولهذا الغرض يعود تحليل النظم المرتب إلى أساليب عديدة معروفة جيداً على مستوى التفاصيل: أشجار القرارات وجداول القرارات والإنجليزية المرتبة والإنجليزية المضبوطة. وإحدى المشاكل التي تظهر من اختبار أي من هذه الأساليب هو أنه كلما كان الأسلوب أكثر رسمية مثل جداول القرارات والإنجليزية المرتبة كلما كان من الممكن تحويله إلى شفرة بواسطة المبرمجين لكنه يكون أكثر صعوبة للمستفيدين غير الملمين بهذه الأنشطة بالنسبة للتغذية المرتجعة. وعلى العكس كلما كانت الوسيلة أقل رسمية مثل أشجار القرارات والإنجليزية المضبوطة كلما جعل ذلك تغذية المستفيد المرتجعة أكثر إمكانية لكنه يتطلب عملاً أكثر بالنسبة



شكل 10.5

تحليل النظم المرتب: تفاصيل صندوق «عملية الحجز»

للمبرمجين. وأحد المحاولات التي يجريها محلل النظم لاستخدام طريقتين لصندوق عملية على أدنى مستوى يظهر لسوء الحظ سؤالاً عن إمكانية تكامل المشكلة.

ومخازن البيانات هي تجميعات لحقول من المعلومات. ويمكن تمييز ما ينتمي إليه أحد الحقول الموجود في أحد المخازن من دراسة سريان البيانات من المخزن وإليه. وهناك سؤال آخر عن الحقول الموجودة في مخزن البيانات وهو الطريقة التي

يجب أن تمر بها الى مصمم الملف أو مصمم قاعدة البيانات بعد مرحلة تحليل النظم. والفلسفة المحددة لتحليل النظم المرتب هو أن الحقول الموجودة في مخزن البيانات يجب أن تمر في صورة تكون مجمعة فيها على هيئة فئات مبسطة (والتي يمكن أن تحتوي على بيانات غير متكررة إذا نفذت مباشرة كملفات واقعية). وهذه تعتبر الصيغة الأوضح للمصممين لاستخدامها في مرحلة التصميم. والأسلوب المقترح لتحقيق ذلك هو تطبيع البيانات Data normalization. وتطبيع البيانات عادة ما يصاحب تصميم قاعدة البيانات على عكس تحليل النظم (وسوف يناقش بالتفصيل). وهناك حجتين خاصتين بتطبيع البيانات وتستخدمان كجزء من تحليل النظم المرتب. أحد الأشياء هو أن تطبيع البيانات يتطلب فهماً دقيقاً للمعنى اللغوي الحقيقي للحقول وللعلاقات بين الحقول بعضها البعض في التطبيق ويكون محلل النظم من يكون له الفهم الأعمق والأسبق لذلك. بالإضافة إلى ذلك فإن أحد الأساليب النمطية لتصميم قواعد البيانات يتطلب تطبيع البيانات كأول خطوة له بغض النظر عن التكوين الواقعي المستخدم. وعلى هذا فإذا كان التطبيع سينفذ بدون أي شروط فيجب أن ينفذ عند هذه المرحلة بالرغم من أن الوسيلة الواقعية لم يتم اختيارها حتى الآن.

وحيث ان كل رسومات مسارات البيانات DFD والمعلومات المصاحبة لها يجب أن تسجل وتحفظ فهناك اقتراح لطريقة التسجيل. وهذه الطريقة هي استخدام قاموس بيانات (سيناقش بالتفصيل في الفصل 11). والطريقة تملي أن يكون محللي النظم هم أصل العناصر الموجودة في القاموس. ويشمل ذلك صناديق العمليات ومسارات البيانات ومخازن البيانات والكينونات الخارجية وأوصاف الحقول كذلك. ويفترض أن تستخدم الأخيرة وتتسع بواسطة العاملين في التصميم الواقعي في وقت لاحق.

هناك وجه آخر لتحليل النظم المرتب وهو عمله كسطح بيني لأساليب تصميم النظم المرتبة. وكما سبق ذكره فكل صندوق لعملية على المستوى الأدنى يميل إلى أن يصبح جزءاً من الشفرة في مفهوم التصميم المرتب. وعلى هذا فيتوقع أن يحول مصمم البرنامج صناديق عمليات أدنى مستوى في DFD (والمنطق الشامل) الى تصميم أجزاء من القمة الى القاعدة. ومن الواضح أن عمل مصمم البرنامج لم ينتهي فليس عليه أن يقوم بعملية التحويل المذكورة فقط ولكن

بالإضافة إلى ذلك يمكن أن تدمج الوظائف الموجودة في بعض صناديق العمليات في جزء واحد من أجزاء البرنامج أو كبديل لذلك يمكن أن تجزء الوظيفة الموجودة في أحد صناديق العمليات إلى عدة أجزاء من أجزاء البرنامج وكل ذلك لتحقيق واقعية النظام الطبيعي.

يمكن ملاحظة العديد من الأشياء بالنسبة إلى استخدام أساليب تحليل النظم المرتب وذلك بالنسبة الى مراحل تطوير النظم النمطية. والأكثر معنوية في ذلك هو أنه يجب على محلل النظم أن يعد DFD أولاً للنظام الموجود (إذا لم يوجد واحد بالفعل سبق إعداد النظام منه). وبعد ذلك ومع تقدم مرحلة التحليل والحصول على موافقة الإدارة يجب عليه أن يعد DFD للنظام المقترح. وهناك نقطة أخرى وهي أن DFD يمكن تجزئته ليعكس أن أحد التنفيذات الطبيعية بعينه يمكن أن يشمل جزءا من التطبيق الموجود تحت الدراسة وذلك لأسباب اقتصادية أو لأسباب أخرى. وهناك عنصر ثالث وهو أن المعلومات الإضافية يمكن أن توضع داخل DFD حتى بعد انتهاء مرحلة التحليل وذلك على صورة ملاحظات في صناديق العمليات التي تحدد مكان وكيفية تنفيذ الوظيفة.

الحاجة إلى تصميم قاعدة البيانات :

The Need for Database Design

هناك أسباب مختلفة تجعل من تصميم قاعدة البيانات ضرورة. ويشمل ذلك تكرار البيانات وأداء التطبيق واستقلالية البيانات وأمن البيانات وسهولة البرمجة. وكل هذه عوامل هامة في بيئة تشغيل البيانات وكلها يمكن أن تتأثر بالتصميم الضعيف لقاعدة البيانات.

تكرار الملفات المتعددة ومحتويات الملفات :

تصميم قاعدة البيانات المناسب ضروري في تجنب كل من التكرارات المتعددة والمتداخلة التي سبق مناقشتها. والكلام الواضح يحدد أن كمية التكرار الواضحة (للحقول الرئيسية وغير الرئيسية) متغير مع تغير نوع نظام إدارة قاعدة البيانات التي تصمم لها الملفات. وفي كل الحالات وعلى الأقل في المراحل المبكرة للتصميم يكون الهدف هو حذف كل التكرار من الحقول غير الرئيسية. وفي

المراحل المتأخرة من التصميم ومع أخذ البيانات شكل تكوين نظام إدارة قاعدة بيانات محدد قد يكون أو لا يكون هناك تكرار صريح في الحقول الرئيسية. وبصفة خاصة يتطلب المنهج العلاقي ظهور حقول معينة كخواص لعلاقات مختلفة بينها في حالات الإبحار يتم استبدال هذا الإزدواج بواسطة المشيرات.

الأداء:

يمكن أن يتأثر الأداء وهو سرعة تشغيل التطبيقات والنظم بعدد من العوامل في بيئة قاعدة البيانات. والعديد من هذه العوامل هي أمور مشتركة في تشغيل البيانات ولا تقع في مدى تحكم قاعدة البيانات. والعناصر الأخيرة تشمل سرعة تشغيل وحدة التشغيل المركزية ومعدل نقل البيانات من القرص وسرعة القناة واتصال التطبيقات المختلفة التي تشترك في استخدام نفس نظم المكونات وبالطبع كفاءة الأساليب الموجودة في نظام إدارة قاعدة البيانات نفسه.

وعلى أية حال هناك العديد من العناصر في بيئة قاعدة البيانات التي تؤثر على الأداء وتقع تحت تحكم مصمم قاعدة البيانات ومعظمها خاصة بالمنتج أي أنها تعتمد على صفات نظام إدارة قاعدة البيانات الخاصة. إلا أن بعضها الآخر عام بدرجة كافية في طبيعته تجعله يستحق الذكر هنا مع ذكر تفاصيله فيما بعد. ومعظمها مرتبط مباشرة أو بصورة غير مباشرة بتخزين أداء قاعدة البيانات: عدد الاتصالات وطبيعتها للبيانات في الاتصال المباشر ووحدات التخزين الثانوية والمعروفة بأنها وحدات مدخلات ومخرجات الأقراص.

وأحد الأمثلة الشائعة للاختيار المرتبط بالأداء هو، أي طريقة اتصال تستخدم وكيفية تنفيذها بالضبط بالنسبة للتطبيق المعني. فقد يقرر المصمم أن يستخدم طريقة اتصال متكرر أو طريقة مفهرسة فما هو عدد الحقول في الفهرس وأي وضع للبيانات أو أي طريقة تكرارية تستخدم (في حالة الطريقة المتكررة) وما إلى ذلك. ويمكن أن يوجد العديد من أنواع المشيرات لتصل السجلات المختلفة مع بعضها البعض كل منها بأداء متشعب لأنواع مختلفة من التطبيقات. ويمكن أن توجد طرق مختلفة لتصميم وضع أنواع مختلفة للسجلات بجانب بعضها البعض في هرمية أو في شبكة. وبالإضافة إلى ذلك هناك بدائل لتجميع حدوثات على نفس وحدة الاتصال المباشر. وسوف توجد توليفات أكثر على نظم محددة فيما بعد.

استقلالية البيانات:

استقلالية البيانات، وهي المقدرة على تعديل تكوينات البيانات دون التأثير على البرامج الموجودة بالفعل، هي وظيفة أساسية لنموذج البيانات الذي يستخدمه نظام إدارة قاعدة بيانات معين. كل هذه النظم تقدم درجة موسعة من استقلالية البيانات عند مقارنتها بتشغيل الملفات المعتاد. وبالرغم من ذلك هناك اختيارات لتصميم قاعدة البيانات يمكن عملها بصفة خاصة عند التعامل مع نظم إدارة قواعد البيانات الهرمية والشبكية والتي يمكن أن تؤثر على درجة الاستقلالية. ودرجة تعقيد الموضوع لا تنتهي عند ذلك لأنه بعد مواصفات تكوين البيانات وتصميم قاعدة البيانات يمكن أن تؤثر طريقة كتابة البرنامج على استقلالية البيانات.

أمن البيانات:

يوجد في نظم إدارة قاعدة البيانات بصفة عامة وقايات مبنية داخلياً لأمن البيانات. وتتراوح هذه من كلمات مرور تصاحب المستفيد إلى كلمات مرور تصاحب كل بيانات محددة الى الطرق المختلفة لمنع الكل باستثناء بعض المستفيدين في أداء عمليات معينة على البيانات. ويصبح أمن البيانات موضوعاً في تصميم قاعدة البيانات عندما تكون أقل كمية من البيانات التي يكون النظام قادراً على إعادتها من قاعدة البيانات طبقاً لاستدعاء البرنامج أكبر من الكمية المسموح للشخص الذي ينفذ البرنامج برؤيتها. وفي هذه الحالة قد يكون تم تصميم الهيكل بحيث أن البيانات، والتي يمكن في الحالات الأخرى أن توضع مع بعضها في وحدة يمكن استرجاعها منها، تقسم إلى العديد من الوحدات.

سهولة البرمجة:

بينما يتطلب استخدام نظم إدارة قاعدة البيانات أن يتعلم المبرمجون مفاهيم واتفاقيات جديدة فإن التأثير الكلي لاستخدام نظم إدارة قواعد البيانات هو تقليل تعقيد البرمجة ومعدلات أخطاء البرمجة. أحد أسباب ذلك هو النمطية التي تنتج من استخدام نفس الاتفاقيات لمعالجة البيانات في كل البرامج. ويشمل سبب آخر الطبيعة المتكاملة لتكوينات البيانات والتي تسمح للمبرمج بالحصول على بيانات

من ملفات مستقلة واقعياً عن بعضها أو من أنواع سجلات مميزة (طبقاً لنموذج البيانات المستخدم) باستدعاء واحد من سطح بيني مرتفع المستوى للبرمجة. وبدون استخدام نظم إدارة قواعد البيانات غير المقبولة فيكون البديل هو الاتصال بالبيانات في ملف واحد واستخدامها في الاتصال ببيانات في ملف آخر وهكذا خلال العديد من الملفات اللازمة لتلبية الاستفسار. هذه الطريقة الأخيرة ينتج عنها أخطاء كثيرة دون ذكر استغلال وقت أطول في الكتابة.

وداخل المستوى المعدل للبرمجة الذي يقدمه استخدام نظم إدارة قواعد البيانات يمكن لأخذ قرارات معينة في تصميم قواعد البيانات أن يجعل من عمل المبرمج عملاً أصعب أو عملاً أسهل. ولسوء الحظ فإن القرارات التي يمكن أن تبسط من مواصفات البرمجة عادة ما يكون لها تأثير عكسي على تكرارية البيانات أو أداء النظام أو على كلاهما. مثال ذلك دمج حقول مع بعضها داخل وحدة تكوين بيانات واحدة قد يسمح للمبرمج أن يبسط المواصفات اللازمة لاسترجاع هذه البيانات إلا أنه عادة ما ينتج عنه زيادة في تكرار البيانات. وإضافة الفهارس الإضافية في تكوين البيانات ينتج عنه مرونة أكثر في الاسترجاع في تشغيل الخط المفتوح مع تبسيط مواصفات استرجاع البرنامج وذلك على حساب الأداء عندما يجب تحديث البيانات فالملف مرتفع التقلبية مع عدد كبير من الفهارس المتصلة يمكن يقلل من أداء النظام الداخلي.

طرق تصميم قواعد البيانات: Database Design Methodologies

يقدم هذا القسم أسلوبين من الأساليب الأكثر شيوعاً لتصميم قواعد البيانات. وفي كل من الحالتين نبدأ بمعلومات موجهة ناحية البيانات يتم انتاجها كمخرجات من تحليل النظم ونستمر في نماذج قواعد البيانات العلاقية وشبه العلاقية والشبكية والهرمية. وطريقة التوضيح هي باستخدام مثال مع التعليقات المناسبة على طول الطريق.

الطريقة الأولى: هي تطبيع البيانات وعمل تكوينات Data normalization and data structuring تمثل فئة من الطرق التي تأخذ قائمة من الحقول والمصاحبات خلالها كمدخلات. والطريقة الثانية: وهي طريقة العلاقة والكينونة

entity - relationship تمثل فئة من الطرق تأخذ الكينونات والعلاقات كمدخلات.

تطبيع البيانات وعمل تكوينات البيانات:

المثال: الطريقة الأولى لتصميم قاعدة البيانات تتطلب خطوة رئيسية للمنهج العلاقي وخطوتين رئيسيتين من الثلاث مناهج الأخرى. ولحسن الحظ وللإقناع في التوضيح فإن الخطوة الرئيسية من المنهج العلاقي هي في الواقع أول خطوة من الخطوتين المأخوذتين من الثلاثة مناهج الأخرى.

ويبين شكل 10.6 مثالاً لمجموعة حقول تهم نظام معلومات الأفراد الخاص بورش إصلاح السيارات على مستوى الدولة. ومن الواضح من هذه القائمة بالحقول أن هدف النظام هو حفظ تتبع التقنيين في الشركة ومهاراتهم ومواقعهم ومعلومات أخرى مصاحبة لهم. ويهدف الحقل المسمى بمستوى المهارة التقنية للميكانيكي MECHANIC'S SKILL PROFICIENCY LEVEL إلى تخزين مستوى مهارة تقني معين. (ويستحق الكلام جانبياً هنا عن ملاحظة أن مجموعة الحقول هذه قد اختيرت للمثال لعمل بعض النقاط عن تصميم قاعدة البيانات. وكما سوف نرى فالنسبة المئوية المرتفعة للحقول في هذا التجميع سوف تسهم كحقول رئيسية عما هو الحال في معظم الحالات الواقعية). ويبين شكل 10.7 المصاحبات عبر حقول هذا المثال. الأسهم التي لها رأس واحدة تحدد تعريفات فردية. وعلى هذا فالمصاحب رقم 2 يحدد أنه لرقم ميكانيكي معين يوجد اسم واحد بالضبط مصاحب له (رقم الميكانيكي يكون فردي). والمصاحب رقم 10 يحدد أنه لرقم ورشة معينة يوجد مدينة واحدة بالضبط مصاحبة لها: فإذا أعطيتك رقم الميكانيكي 81 فيمكنك أن تعطيني بصورة فردية John Smith. وإذا أعطيتك رقم الورشة 21 فيمكنك أن ترد على بصورة فردية بالرد Chicago. لاحظ أنه بدون التحديد الصريح لا يمكن افتراض الرد أن العكس صحيح. وعلى هذا فيمكن أن يوجد في الشركة العديد من John Smith كما يمكن أن يوجد العديد من الورش في Chicago. والأسهم ذات الرؤوس المزدوجة في شكل 10.7 تحدد المصاحبات

المتعددة: مثل العنصر الموجود على اليسار والذي يمكن أن يصاحبه العديد من العناصر على اليمين. المصاحب رقم 1 يقول أن الميكانيكي يمكن أن يكون له

MECHANIC NUMBER (MECH NO)
SKILL NUMBER (SKLL NO)
SKILL CATEGORY (SKLL CAT)
MECHANIC NAME (MECH NAME)
MECHANIC AGE (MECH AGE)
SHOP NUMBER (SHOP NO)
SHOP CITY
SHOP SUPERVISOR (SUPV)
MECHANIC'S SKILL PROFICIENCY LEVEL (PROF)

شكل 10.6
مثال لفئة من الحقول

1. MECH NO ——→→ SKLL NO
2. MECH NO ——→ MECH NAME
3. MECH NO ——→ MECH AGE
4. MECH NO ——→ SHOP NO
5. MECH NO ——→ SUPV
6. SKLL NO ——→→ MECH NO
7. SKLL NO ——→ SKLL CAT
8. SHOP NO ——→→ MECH NO
9. SHOP NO ——→ SUPV
10. SHOP NO ——→ SHOP CITY
11. MECH NO, SKLL NO ——→ PROF

شكل 10.7
المصاحبات بين الحقول

مهارات متعددة. والمصاحب رقم 8 يقول أن الورشة يمكن أن يكون بها العديد من الميكانيكين الذين يعملون بها.

مشاكل رئيسية في قواعد البيانات: قبل البدء في عملية تصميم قاعدة البيانات يكون من المفيد إعادة تعريف المشاكل بأن التصميم الرديء لقاعدة البيانات يمكن أن يحدث في محتوى هذا المثال. يبين شكل 10.8 سبعة سجلات عينه لملف أعد من كل الحقول التي سردت في شكل 10.6. ويفترض أنه لم يبذل أي مجهود تصميم لقاعدة البيانات. وعلى هذا فكل الحقول تكدست مع بعضها في

Key: MECH NO	Key: SKLL NO	SKLL CAT	MECH NAME	MECH AGE	SHOP NO	SHOP CITY	SUPV	PROF
21	113	Body	Adams	55	52	NYC	Brown	3
35	113	Body	Baker	32	44	LA	Green	5
35	179	Engn	Baker	32	44	LA	Green	1
35	204	Tran	Baker	32	44	LA	Green	6
50	179	Engn	Cody	40	44	LA	Green	2
77	148	Tire	Doe	47	52	NYC	Brown	6
77	361	Engn	Doe	47	52	NYC	Brown	6
. . .								

شكل 10.8
عينة لسجلات

ملف واحد. وقد اختير MECHANIC NUMBER و SKILL NUMBER كمفتاح مركب (حقل متعدد) حيث أن أخذ الاثنين معاً يمكن أن يعرف بصورة فريدة سجل من السجلات الموجودة في الملف. لاحظ أن ذلك يحدث حيث أن القيمة الرئيسية يمكن أن تعرف بصورة فريدة سجلاً من سجلات الملف والقيمة الرئيسية لهذا السجل (وبالعكس فإن القيم غير الرئيسية تكون معتمدة على القيمة الرئيسة).

من الواضح أنه هناك كم هائل من التكرار في الملف في شكل 10.8. فالميكانيكي رقم 35 له ثلاثة مهارات. وتمثيل هذا يتطلب 3 سجلات حيث يجب أن تؤخذ MECHANIC NUMBER و SKILL NUMBER معاً لتكون حقلاً رئيسياً مقبولاً. في الواقع كل زوج MECHANIC NUMBER و SKILL NUMBER يجب أن يظهر في سجل منفصل لكي يكون مصاحباً مع رقم PRO-FICIENCY LEVEL والذي يكون معتمداً على الخليط من الميكانيكي المطلوب والمهارات الموجودة تحت المناقشة. ونتيجة لهذا الموقف فإن أحد أمثلة التكرار يجب أن تحدث بظهور عمر واسم الميكانيكي ورقم الورشة ومدينة الورشة والمشرف عليه عدد من المرات يساوي نفس عدد مهارات الميكانيكي. وهناك مثال آخر وهو المعلومات عن من هو المشرف في ورشة معينة الذي يجب أن يظهر على الأقل مرة واحدة مع كل ميكانيكي يعمل في هذه الورشة. وتحضر هذه الحقائق مع نفسها كل المشاكل النمطية للتكرار: مساحات تخزين أكثر ووقت أكثر للتحديث ومشاكل سلامة كبيرة.

وهناك مشكلة أساسية أخرى مع الملف وتشمل حذف وإضافة البيانات. إفرض أن Adams و Doe هما الميكانيكيين الوحيدين في الورشة رقم 52. فإذا أنهيا عملهما وكما هو متوقع فإننا نحذف السجلات الخاصة بهما وبهذا فإننا نكون قد حذفنا معهما الحقيقة بأن Brown المشرف عليهما في الورشة 52 من الملف دون أن يكون قد أنهى عمله بالورشة. هاتين القطعتين الأساسيتين من المعلومات لا يمكن أن توجدا منفصلتين في التصميم الحالي. ويوجد موقف شبيه إلا أنه عكسي في إضافة البيانات للملف: فمن المعتاد إذا أضفنا ميكانيكي جديد للملف فإننا

نتوقع أن نجد معلومات عن مدينة الورشة وعن المشرف من السجلات الموجودة بالفعل في الملف والخاصة بهذه الورشة.

أما إذا كان الميكانيكي الجديد هو أول ميكانيكي يعمل بالورشة فعند ذلك لا توجد معلومات عن مدينة الورشة أو المشرف في الملف ويجب البحث عنه في مكان آخر. وإضافة ميكانيكي ليس لديه أية مهارات يمكن أن يكون له تأثيرات جادة بالنسبة للقيمة الرئيسية. وعلى هذا فمن الواضح أن وضع كل الحقول ببساطة في ملف واحد يترك كماً كبيراً من المشاكل.

تطبيع البيانات: تطبيع البيانات هي طريقة لترتيب الحقول داخل جداول (أو ملفات أو علاقات) بحيث أن التكرار عبر الحقول غير الرئيسية يختفي. وكل من الجداول الناتجة يتعامل مع منطقة فردية من المعرفة على عكس الخليط المبين في شكل 10.8. المدخلات اللازمة لعملية التطبيع هي قائمة بفئة حقول البيانات والمصاحبات لها كما هو موجود على سبيل المثال في شكلي 10.6 و 10.7. وبالرغم من ذلك فإنه يكون من الأسهل توضيح العملية في محتوى مثال يشمل سجلات كعينه وعلى هذا فسوف نستمر بالمثال الذي سبق لنا استخدامه. نؤكد مرة أخرى أن قوائم حقول البيانات ومصاحباتها وهي المدخلات اللازمة هنا هي أجزاء من مخرجات مرحلة تحليل النظم.

البيانات غير المطبعة: يبين شكل 10.9 بيانات المثال في ترتيب تكويني غير كامل بعض الشيء يسمى بصورة غير المطبعة unnormalized. ويمكن أن تشمل البيانات غير المطبعة مواقف مثل حقول لها قيم متعددة. مثال ذلك السجل الثاني في شكل 10.9 والذي يشير إلى الميكانيكي رقم 35 والملاحظات داخل السجل الفردي بأنه لديه ثلاثة مهارات. هذه الصيغة للبيانات لها بعض العيوب الميزة بما في ذلك الحاجة إلى شكل معين للطول المتغير للسجلات والتعقيد الإضافي للمبرمج والذي يمكن أن يقود إلى زيادة في أخطاء البرمجة. والأكثر من ذلك ففي حالة سجل الميكانيكي رقم 35 فإننا نعرف أن المهارة رقم 113 تتفق مع فئة المهارة الخاصة بالجسم body وهكذا خلال كل الأزواج المسرودة لأرقام المهارات والفئات

MECH NO	SKLL NO	SKLL CAT	MECH NAME	MECH AGE	SHOP NO	SHOP CITY	SUPV	PROF
21	113	Body	Adams	55	52	NYC	Brown	3
35	113 179 204	Body Engn Tran	Baker	32	44	LA	Green	5 1 6
50	179	Engn	Cody	40	44	LA	Green	2
77	148 361	Tire Engn	Doe	47	52	NYC	Brown	6 6

شكل 10.9
بيانات غير مطبعة

Key (MECH NO, SKLL NO)

MECH NO	SKLL NO	SKLL CAT	MECH NAME	MECH AGE	SHOP NO	SHOP CITY	SUPV	PROF
21	113	Body	Adams	55	52	NYC	Brown	3
35	113	Body	Baker	32	44	LA	Green	5
35	179	Engn	Baker	32	44	LA	Green	1
35	204	Tran	Baker	32	44	LA	Green	6
50	179	Engn	Cody	40	44	LA	Green	2
77	148	Tire	Doe	47	52	NYC	Brown	6
77	361	Engn	Doe	47	52	NYC	Brown	6

شكل 10.10
الصورة الطبيعية الأولى

إلا أن هذا الاتفاق قد لا يوجد بصفة دائمة في كل المواقف الشبيهة متسبباً في أحداث اضطراب.

الصورة الطبيعية الأولى: بيانات الصورة الطبيعية الأولى لها خاصية أن كل محتوى بيانات أو قيمة حقل يجب ألا يكون قابلاً للتحليل. وشكل 10.10 هو تمثيل للصورة الطبيعية الأولى لبيانات شكل 10.8. فكل محتوى حقل لكل سجل يحتوي على قطعة واحدة من بيانات غير قابلة للتجزئة. وبالضرورة فإن السجلات غير المطبعة داخل الحقول التي لها قيم متعددة تنتج سجلات عديدة ببعض بيانات متكررة كلما دعت الحاجة لذلك. فيما عدى للحقيقة بأن البيانات الموضحة في شكل 10.8 تمثل جزءاً واحداً فقط من الملف بينما بيانات شكل 10.10 تمثل محتويات الملف (بغرض التوضيح) والاثنين متطابقين. ومن الواضح أن تمثيل الصورة الطبيعية الأولى للبيانات ليس مساعداً كترتيب لمراقبة التكرار في حد ذاته إلا أنه يعتبر نقطة للقفز للمزيد من العمل.

الصورة الطبيعية الثانية: لقد حددنا بالفعل أن البيانات الموجودة في شكل 10.10 (مثل شكل 10.8) متكررة بصورة مرتفعة. عند هذه النقطة تعود الطريقة إلى السؤال عما ينظر إليه لتغييره في تكوين البيانات وذلك لبدأ تخفيف التكرار.

الحقيقة أن خليط الحقول MECHANIC NUMBER - SKILL NUMBER هي مفتاح رئيسي صحيح لهذا الملف تم إقرارها بالفعل. مرة أخرى يعني ذلك من ضمن أشياء أخرى إن كل حقل غير رئيسي في الملف يعتمد على الحقل الرئيسي وهذا يمكن التأكد من صحته بالمصاحبات ذات السهم ذو الرأس الواحدة في شكل 10.7. بينما كل من جزئي المفتاح الرئيسي المركب ضروري لتعريف حقل PROFICIENCY FIELD (المصاحب رقم 11 في شكل 10.7) فلا يكون هناك حاجة إلا إلى واحد فقط في هذين الجزئين لتعريف كل من الحقول الأخرى غير الرئيسية. هذه الحقيقة هي مفتاح التكرار وحذفه ولو جزئياً.

يبين شكل 10.11 نفس البيانات في الصورة الطبيعية. وقد قسمت إلى ثلاثة جداول كل منها له خاصية أن حقله الرئيسي الداخلي مطلوب لتعريف كل حقل

من حقوله غير الرئيسية. وداخل أحد الجداول المعطاة لا يعرف أي حقل غير رئيسي بواسطة جزء من الحقل الرئيسي فقط. وقد تم عمل ازدواج للعديد من الحقول في هذه العملية MECHANIC NUMBER والذي يظهر بالطبع مرة واحدة فقط كحقل في الصورة الطبيعية الأولى يظهر الآن في كل من جدول Mechani وجدول Proficiency. وكما سوف نرى فإن هذا النوع من ازدواج الحقول ضروري عند هذه المرحلة (وبالنسبة للحالة العلاقية بصفة عامة) عبر الحقول التي تعمل كحقول رئيسية. وفي الواقع فإن إجمالي عدد حدوثات قيم الحقل تقل من 63 في شكل 10.10 إلى 55 في شكل 10.11 محدداً تقليلاً في التكرار. لاحظ كذلك أن المناطق المعرفة فردياً للمعرفة تمثل في كل من الجداول (بالرغم من أنها ليست في الدرجة النهائية كما سوف نرى).

الصورة الطبيعية الثالثة: جدول Skill وجدول Proficiency في شكل 10.11 هما جدولان حران تماماً من تكرار الحقول غير الرئيسية عند هذه النقطة. في الواقع كل منهما موجود في الصورة الطبيعية الثالثة فعلاً. ولكن نظرة على جدول Mechanic تظهر أنه هناك بعض التكرار الذي ما زال موجوداً. مثال ذلك كل من السجلين 1 و 4 يحدد أن الورشة رقم 52 تقع في نيويورك ويشرف عليها Brown. ومثل ذلك السجلان 2 و 3 والورشة رقم 44 وطبقاً لقائمة العلاقات فإن الحقل الرئيسي MECHANIC NUMBER يعرف كل من حقوله الأخرى. وحيث أن الحقل الرئيسي يحتوي على حقل واحد فقط فإن قاعدة الصورة الطبيعية الثانية لاعتماد الحقول غير الرئيسية على محتوى الحقل الرئيسي لا تتحقق بالطبع.

والمشكلة هي العامل مع المصاحبات بين SHOP NUMBER و SHOP CITY وبين SHOP NUMBER و SUPERVISOR (المصاحبات رقم 9 ورقم 10 في شكل 10.7). يقوم SHOP NUMBER بتعريف SHOP CITY وSUPERVISOR. لكنه من الواضح أنه ليس مفتاح رئيسي لمحتويات الجدول (فهو لا يعرف MECHANIC NUMBER أو MECHANIC NAME أو MECHANIC AGE). وعلى هذا فالموقف هو موقف لحقل غير رئيسي يعرف حقول غير رئيسية أخرى. هذا النوع من العلاقات الخاصة وهو نوع آخر من

Key

MECH NO	MECH NAME	MECH AGE	SHOP NO	SHOP CITY	SUPV
21	Adams	55	52	NYC	Brown
35	Baker	32	44	LA	Green
50	Cody	40	44	LA	Green
77	Doe	47	52	NYC	Brown

Mechanic table

Key

SKLL NO	SKLL CAT
113	Body
148	Tire
179	Engn
204	Tran
361	Engn

Skill table

Key

MECH NO	SKLL NO	PROF
21	113	3
35	113	5
35	179	1
35	204	6
50	179	2
77	148	6
77	361	6

Proficiency table

شكل 10.11
الصورة الطبيعية الثانية

تحديد أنواع معلومات أساسية مختلفة تختلط مع بعضها في نفس الجدول هو ما يتسبب في التكرار الذي يظل موجوداً بالجدول.

ويبين شكل 10.12 تمثيل الصورة الطبيعية الثالثة للبيانات. لاحظ أنه تم

Key

MECH NO	MECH NAME	MECH AGE	SHOP NO
21	Adams	55	52
35	Baker	32	44
50	Cody	40	44
77	Doe	47	52

Mechanic table

Key

SKLL NO	SKLL CAT
113	Body
148	Tire
179	Engn
204	Tran
361	Engn

Skill table

Key

SHOP NO	SHOP CITY	SUPV
44	LA	Green
52	NYC	Brown

Shop table

Key

MECH NO	SKLL NO	PROF
21	113	3
35	113	5
35	179	1
35	204	6
50	179	2
77	148	6
77	361	6

Proficiency table

شكل 10.12

الصورة الطبيعية الثالثة

إنتاج جدول جديد وهو جدول Shop وذلك لفصل البيانات المرتبطة بالورشة. لاحظ كذلك أنه في تجزئة جدول Shop تركت نسخة من عمود SHOP

NUMBER خلف جدول Mechanic . وقد حدث ذلك لأنها كانت هذه هي الطريقة الوحيدة للاستمرار في تحديد أي ورشة يصاحبها الميكانيكي .

وعلى هذا ففي الصورة الطبيعية الثالثة نستطيع أن نقول أنه في كل جدول لا يوجد موقف يعرف فيه حقل غير رئيسي حقلاً غير رئيسياً آخراً. وهناك استثناء مشترك لهذا عندما يكون هناك حقلان يمثلان الحقل الرئيسي مثل رقم تعريف الموظف ورقم الضمان الاجتماعي . فالحقل الذي لم يحدث له اختيار من الحقلين كحقل رئيسي يبدو على أنه حقل غير رئيسي وهو يعرف على ذلك حقولاً غير رئيسية أخرى ولكن في هذه الحالة الخاصة لا يوجد كسر للقاعدة المحددة. وتحتوى بيانات الحالة الطبيعية الثالثة الموجودة في شكل 10.12 على عدم تكرار عبر الحقول غير الرئيسية. وهذا هو الهدف الذي سعينا له .

صور طبيعية أخرى: مع استمرار البحث أكثر في هذا المجال تظهر مشاكل أخرى وحلول لها. وقد سبق تعريف استثنائين يشملان الصورة الطبيعية الثالثة للبيانات والتي ما تزال تحتوي على تكرار. كل من الاستثنائين يحتوي على خليط من المصاحبات المتعددة. وقد سميت بالصور الطبيعية الرابعة والخامسة وتظهر بصورة غير متكررة عملياً وتقع خارج نطاق هذا الكتاب.

قيمة الجداول التي يحدث لها تطبيع : يمكن عمل ملاحظتين رئيسيتين عن تطبيع البيانات. إحداهما هي أن هذه الصورة للبيانات الخالية من التكرار بالنسبة للحقول غير الرئيسية هي مستوى وسيط في تصميم تكوينات البيانات لقواعد البيانات الهرمية والشبكية.

* النقطة الثانية هي أن تطبيع البيانات المرتبط بالتعديلات التكوينية الممكنة لأسباب الأداء هو التصميم النهائي لقواعد البيانات. وفي الواقع تؤكد عملية التطبيع أن حقول الوصل تظهر في كل الجداول التي يجب أن تظهر فيها طبقاً للعلاقات المعطاة بين الحقول. وهذه ملاحظة قوية جداً وحقيقة مركزية لاستخدام التطبيع في تصميم كل أنواع قواعد البيانات.

نفس الملاحظات عن التطبيع الذي ينتج عنه تصميم قاعدة البيانات التى ترتبط بتعديلات الأداء تطبق على قواعد البيانات شبه العلاقية المرتبطة بإضافة الجداول أو الفهارس والتي تصل مسبقاً الجداول المطبعة المبنية على علاقات

البيانات واحتياجات التطبيق. في الواقع نفس الملاحظة الأساسية تطبق كذلك على الملفات المسطحة المعتادة غير الموجودة في قاعدة بيانات والتي يكون التطبيع أسلوب تصميم مثالياً لها.

عمل تكوين البيانات ـ قواعد البيانات الهرمية: بصفة عامة عند التحويل من مجموعة جداول حدث لها تطبيع إلى تكوين بيانات هرمي تتحول هذه الجداول إلى نقاط في الهرمية. وتتصل الحدوثات من نفس أنواع السجل ومن أنواع مختلفة للسجلات بواسطة أنواع مشيرات مختلفة. وفي تأثير ذلك يتم اختيار بعض هذه الجداول ليعد وصلاً مسبقاً لسجلاتها مع بعضها وذلك لتحقيق مجموعة مختارة من العلاقات بصورة طبيعية. والميزة الأساسية لهذا الأسلوب هي الأداء الرائع أثناء تنفيذ الاتصال المبني على هذه العلاقات.

وعملية تصميم قاعدة بيانات هرمية مثل ما هو لنظام DL/I تبدأ بمجموعة من الجداول التي حدث لها تطبيع. وسوف نستمر في تتبع المثال الذي بدأنا استخدامه مع استخدام الجداول التي حدث لها تطبيع والموجودة في شكل 10.12.

عملية إعداد التكوين. تبدأ العملية بملاحظة المصاحبات المتعددة الموجودة في شكل 10.7 وهي القائمة الأصلية للحقول. والمصاحب المتعدد الذي لا يظهر مصاحبة المتعدد المعكوس في القائمة يسجل على أنه علاقة من واحد إلى العديد. وهناك علاقة واحدة من هذا النوع في شكل 10.7

SHOP NO ———>> MECH NO

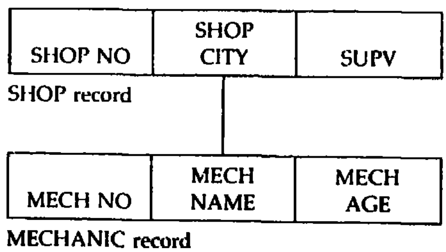
والتي لها رقم مصاحبة 8. أما المصاحب المتعدد الذي يظهر معكوسه في القائمة فيسجل على أنه علاقة من العديد إلى العديد. وهناك علاقة واحدة من هذا النوع في شكل 10.7.

MECH NO <<———>> SKLL NO

والتي تتكون من خليط رقم المصاحبة 1 ورقم المصاحبة 6.

بعد ذلك يتم تشكيل الهرميات من علاقات من واحد إلى العديد. حيث أن كل فرع من أفرع تكوين البيانات يمثل علاقة من واحد إلى العديد فتكون هذه

الخطوة مباشرة. يبين شكل 10.13 هرمية أعدت من علاقة من واحد إلى العديد في المثال. نقاط MECHANIC و SHOP في الهرمية إستخلصتا مباشرة من جدولي Mechanic و Shop الذي حدث لهما تطبيع الموجودين في شكل 10.12. لاحظ أن حقل SHOP NO في جدول Mechanic في شكل 10.12 لم يظهر في سجل MECHANIC في شكل 10.13. وفي الجداول التي حدث لها تطبيع فإن الطريقة الوحيدة لتحديد أي الورش التي يعمل بها الميكانيكي هي تعريف SHOP NO كحقل في جدول Mechanic. ومعنى الحقيقة أن SHOP NO يظهر كذلك كحقل في جدول Shop يبدو أنه صورة من صور التكرار تشمل حقلاً يخدم على الأقل في جدول واحد كحقل رئيسي. ولكن عندما يتم تحويل هذه الجداول إلى سجلات في هرمية تمثل علاقة من واحد إلى العديد بين الورش والميكانيكيين فلا يوجد حقل SHOP NO في سجل Mechanic. والنقطة هي أن معرفة أي ورشة يعمل فيها الميكانيكي تكون مخزنة الآن بواسطة المشيرات التي تصل ورشة معينة بكل الميكانيكيين الذين يعملون بها.

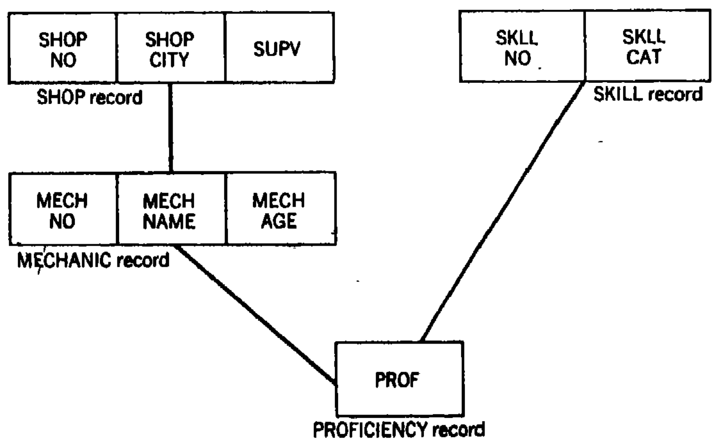


شكل 10.13
هرمية تمثل علاقة من واحد إلى العديد

ولا يمكن بناء علاقة من العديد إلى العديد مباشرة في تكوين هرمي للبيانات. ويجب أن يعد لها محاكاة بواسطة علاقتين من نوع من واحد إلى العديد أو بالعديد من أنواع السجلات التي تنتج سجلاً جديداً بصفة خاصة. وعندما يتم تحميل البيانات باستخدام هذا التكوين يكون هناك حدوث واحد لهذا النوع الخاص من السجلات لكل حدوث للعلاقة (لكل اتصال بين كينونتين في

العلاقة). ويبين شكل 10.14 هرمية شكل 10.13 على اليسار مع هرمية (نوع سجل فردي) على اليمين تحتوي على نوع سجل SKILL. والميكانيكي والمهارات والذي بينهما علاقة من العديد إلى العديد يوجدان في علاقة من واحد إلى العديد مع انتاج نوع سجل بصفة خاصة يسمى PROFICIENCY. نقطة Skill التي تكون الهرمية الجديدة الموجودة على اليمين ثم استخلاصها من جدول Skill الموجود في شكل 10.12. ونقطة Proficiency تم استخلاصها من جدول Proficiency الموجودة في شكل 10.12.

تذكر أن علاقة من العديد إلى العديد التي نفذت تصل حدوثات MECHANIC مع حدوثات SKILL. ولا يحدث جدول Proficiency الموجود في شكل 10.12 هذه التوصيلات فقط لكنه يعطي معلومات إضافية كذلك (مهنية ميكانيكي معين في مهارة معينة) عنها.

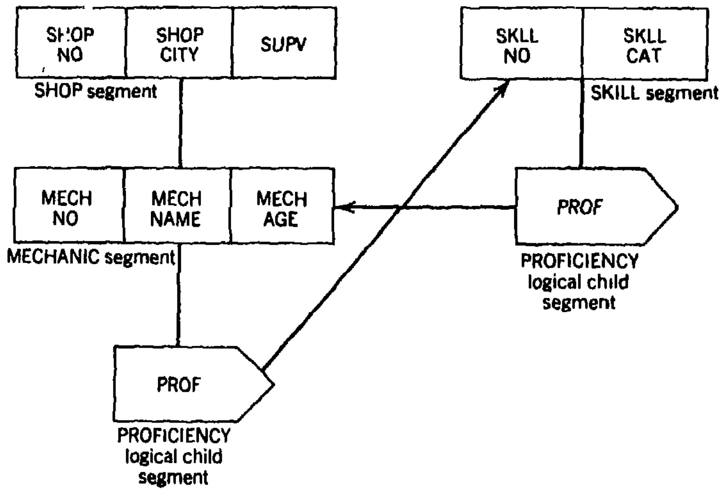


شكل 10.14

هرميتان متصلتان لتمثيل علاقة من العديد إلى العديد

DL/I : يبين شكل 10.15 نفس التكوين الهرمي المبين في شكل 10.14 بقوة فيما عدى أنه حدث تعديل عليه ليتفق مع اصطلاحات DL/I. وقد استبدل اصطلاح سجل record باصطلاح قطاع segment. ونوع السجل الخاص الذي يخدم كواصل لعلاقة من العديد إلى العديد يسمى بالقطاع التابع المنطقي logical

child. والحقيقة التي تظهر في كل من الهرميتين (مع ازدواج بيانات المهنية في كل من المكانين في أحد الصيغ) تحدد أن علاقة من العديد إلى العديد يمكن أن تنفذ في الواقع في كل من الاتجاهين. أي أنه لمهارة معينة فإننا نستطيع إيجاد كل الميكانيكيين الذين يتمتعون بهذه المهارة ولميكانيكي معين نستطيع إيجاد كل المهارات فقط فعند ذلك يمكن أن ترحل العلاقة من هذا الجانب.



شكل 10.15

تمثيل DL/1 يشمل علاقة منطقية

تعديلات الاتصال والأداء: يظل هناك حجم عمل كبير بعد إعداد التطبيع والتكوين ويكون هناك حاجة إلى ضبط التكوين الهرمي التجريبي لأسباب خاصة بالاتصال والأداء والمناسبة والأمن. وهذه العملية هي ما سبق الإشارة إليها بأنها التصميم الطبيعي.

وبالنسبة للاتصال يجب أن يجري المصمم حسابات عن أي الحقول من أي القطاعات يجب أن يحدث لها اتصال بصورة مباشرة. ويكون هذا مبنياً على نتائج تحليل النظم بالنسبة لمجموعة التطبيقات المأخوذة للاستخدام مع قاعدة البيانات. ومن المعتاد أنه يمكن الاتصال مباشرة بقطاعات الجذر فقط بواسطة حقولها الرئيسية فقط عند ذلك (وحتى التوسع في هذه المقدرة هو قرار يجب اتخاذه). لكن إفرض

أن أحد التطبيقات يتطلب اتصالاً مباشراً بحقل في قطاع غير جذري. مثال ذلك أنه إذا كان هناك حاجة في المثال الحالي لإيجاد ميكانيكيين مباشرة بواسطة الميكانيكي. أو إفرض أن التطبيق يتطلب اتصالاً مباشراً بحقل غير رئيسي في قطاع جذري مثال ذلك الحاجة إلى إيجاد ورشة مباشرة عن طريقة المدينة. عند ذلك يجب استخدام الفهارس الثانوية أو يجب إعادة ترتيب التكوين بحيث أن بعض القطاعات غير الجذرية تصبح جذور الهرميات جديدة مع استخدامات علاقات منطقية لتوصيل الهرميات الجديدة والقديمة معاً لحفظ العلاقات المناسبة.

وهناك مثال آخر لقرار يعتمد على التطبيق وهو اختيار طرق الاتصال. فيمكن أن يكون الاتصال بقطاعات الجذر تتابعياً أو مباشراً (مفهرس بصفة خاصة) وتتابعياً معاً أو مباشراً عبر طرق التكرار. وبالطبع مثل هذا القرار يكون مبنياً على الاحتياجات لتطبيقات تستخدم البيانات.

وهناك مدى واسع من العوامل التي يجب أن تؤخذ في الاعتبار في DL/I بالنسبة للأداء والعديد منها مرتبط ببعض: أي أن التغيير في أحدها يمكن أن يتطلب تغيير متوازن أو تعويض في الآخر. مثال ذلك السلاسل الطويلة المزدوجة (مثل عدد كبير من الميكانيكيين الذين يعملون في ورشة واحدة) يمكن أن يكون لها تأثير ضار على الأداء لأن النظام لا يسمح باتصال مباشر في منتصف مثل هذه السلسلة عندما يحدث الاتصال من قطاع أعلى. وبدءاً من بداية السلسلة يمكن أن يحدث الاتصال عن طريق النقل التتابعي فقط لحدوثات القطاعات المتصلة بمشيرات. وطبقاً لطول السلسلة وعدد حدوثات القطاعات التي يمكن إحضارها داخل الذاكرة الرئيسية من القرص في وقت واحد فاجتياز مثل هذه السلسلة يمكن أن يعني عدد من المرات غير المقبولة للاتصالات بالقرص. ومن ضمن حلول هذه المشكلة يوجد إدخال نوع صوري لمستوى قطاع فوق النوع الذي له سلسلة مزدوجة طويلة وزيادة عدد الحدوثات التي يمكن أن تقرأ داخل الذاكرة الأساسية في نفس الوقت.

هناك مثال آخر لاعتبار الأداء يجب أن يؤثر على التصميم ويهم عدد حدوثات قطاع الجذر للهرمية. مرة أخرى يستخدم التكوين الموجود في شكل 10.15 وافتراض أنه هناك عدد بسيط من الورش وعدد وفير من الميكانيكيين في كل

ورشة. وتحت هذه الظروف فإن وجود جذر قطاع SHOP لا يعطي إحساساً كبيراً. والمقدرة التي تقدمها طرق اتصال IMS لامكانية الاتصال بقطاع جذر مباشرة وبسرعة مثل هذا الاتصال باستخدام تسهيلات التنظيم التكراري بصفة خاصة تكون الحالة التي سبق ذكرها يجب أن يصنع قطاع MECHANIC الجذر مع إضافة حقل SHOP NUMBER إلى حقل MECHANIC NUMBER لتكوين مفتاح من حقلين. وينتج عن هذا بعض التكرار في حقل SHOP - NUMBER إلا أنه يعتبر مقبولاً تحت هذه الظروف ويمكن حفظ معلومات SHOP CITY و SUPERVISOR والتي ما تزال غير متكررة في ملف صغير منفصل.

وتشمل الاعتبارات الأخرى للأداء التي تصبح مهمة أكثر أو أقل طبقاً للحالة الخاصة عدد مستويات (عمق) الهرمية والمسافة من جذور قطاعات معينة والأحجام النسبية للقطاعات واستخدام قطاعات متغيرة الأطوال وما إلى ذلك.

عمل تكوين البيانات ـ قواعد البيانات الشبكية: القواعد الأساسية لتصميم قاعدة البيانات باستخدام منهج التطبيع وعمل التكوينات هي نفسها في قواعد بيانات شبكات CODASYL مثل المستخدمة في تصميم هرميات DL/I. مجموعة العلاقات التي يجري لها تطبيع مع قائمة العلاقات من واحد إلى العديد ومن العديد إلى العديد خلال الكينونات هي نقطة البداية.

يبين شكل 10.16 شبكة أعدت من علاقات الصورة الطبيعية الثالثة الموجودة في شكل 10.12 وعلاقة من واحد إلى العديد.

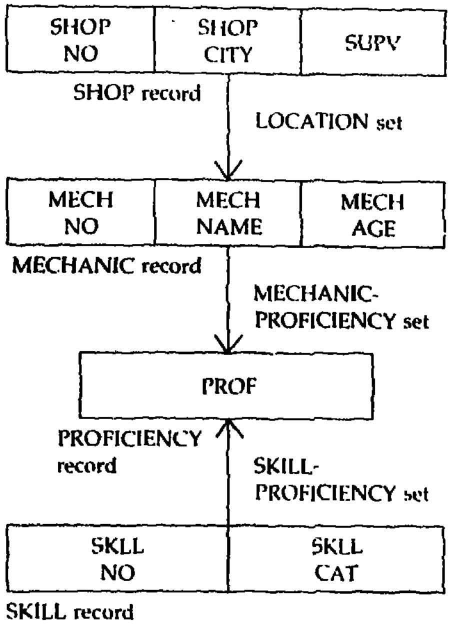
(SHOP NO ——>> MECH NO)

وعلاقة العديد إلى العديد.

(MECH NO <<——>> SKLL NO)

الموجودة في شكل 10.7. لاحظ مواصفات الثلاثة مجموعات المشمولة. لاحظ كذلك السجل الواصل link الذي يصل سجلات MECHANIC و SKILL والذي يحقق علاقة من العديد إلى العديد. وكما هو الحال في إعداد قاعدة بيانات DL/I تماماً الموجودة في شكل 10.15 فقد رسمت علاقة من الواحد

إلى العديد أولاً وتبعتها علاقة من العديد إلى العديد. وفي حالة الشبكة عندما يحين الوقت لرسم علاقة من العديد إلى العديد يضاف سجل SKILL إلى تكوين الشبكة الذي بدء بالفعل بعلاقة من واحد إلى العديد بين سجلات SHOP وMECHANIC. وهذا على العكس ـ التقني ـ لحالة DL/I التي يحدث إعداد هرمية ثانية فيها عندما يراد إضافة سجل (أو قطاع) SKILL إلى الصورة.



شكل 10.16
تمثيل شبكة CODASYL

وللتعديلات الإضافية المبنية على أمور الأداء تنطبق نفس التعليقات المستخدمة مع DL/I في هذه الحالة كذلك. فالعوامل مثل طول السلسلة وأنواع طرق الاتصال وخيارات المشيرات وتعقيدات الشبكة وما إلى ذلك يجب أن تؤخذ في الاعتبار في جو متطلبات التطبيق المحدد ونظام إدارة قاعدة البيانات المستخدم. وعند ذلك يجب أن تجري التعديلات في الشبكة طبقاً لهذه العوامل.

عمل تكوينات ـ البيانات ـ قواعد البيانات شبه العلاقية: مرة أخرى في حالة قواعد البيانات شبه العلاقية فإن العلاقات التي يحدث لها تطبيع وعلاقات من واحد إلى العديد ومن العديد إلى العديد تشمل القفز من نقطة إلى إعداد التكوين. ويحتوي إعداد التكوين على ثلاثة خطوات أساسية.

أولاً : يؤخذ قرار خاص بقابلية تطبيق إنتاج ملفات في الصورة الدقيقة للعلاقات التي حدث لها تطبيع (حيث أن التكوينات الأساسية للبيانات في مثل هذه النظم هو الملفات الخطية البسيطة).

ولا يوجد أي سؤال يحفظ الملفات في نفس الصورة مثل العلاقات التي لم يجري لها تطبيع يكون مثالياً من ناحية تقليل تكرار البيانات. وعلى أية حال ففي لحظات معينة قد تنتج عملية عدم التطبيع عدداً كبيراً من العلاقات تحتوي كل منها على عدد صغير نسبياً من الحقول. ويمكن أن يكون لهذا الموقف تأثيراً سلبياً هاماً على الأداء وعلى تعقيد برمجة التطبيق. وفي هذه الحالة قد يكون من الضروري التسامح مع كمية معينة من التكرارات في دمج الحقول من بعض العلاقات التي لم يجري لها تطبيع لإنتاج ملفات أكبر.

ثانياً : تكوينات البيانات التي تصل ملفات مختلفة مع بعضها البعض يجب أن تصمم. يمكن أن تأخذ مثل هذه التكوينات شكلاً يتفق مع القوائم الخطية أو أشجار الفهارس بناءاً على مفاتيح الملفات المتصلة كما هو الحال في نظام /System 38 من IBM.

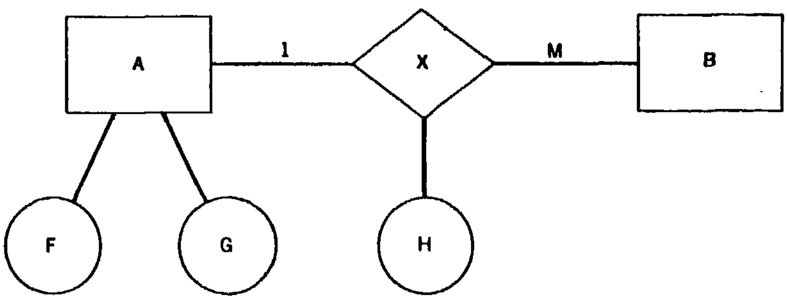
ثالثاً : يجب أن يدرس المصمم متطلبات البيانات لمجموعة التطبيقات ويحدد الحقول لكل ملف معين التي يجب أن يتم الاتصال بها بصورة مباشرة. مثل هذه النظم لها مقدرة على أن يكون لديها بعض أنواع الفهارس معدة لكل حقل يكون له مثل هذه المتطلبات.

نموذج العلاقات والكينونات :

مقدمة : هناك طريقة رئيسية أخرى لتصميم قاعدة البيانات تشمل نموذج علاقات الكينونات. فتصميم قاعدة البيانات باستخدام نموذج علاقات الكينونات يبدأ بقائمة بأنواع الكينونات المشمولة والعلاقات الموجودة بينها. وفلسفة افتراض لأن المصمم يعرف ما هي أنواع الكينونات تختلف بشدة عن الفلسفة من وراء المنهج المبني على التطبيع. وكل من الطريقتين يفترض أنه قد حدث تحليل دقيق للنظم. وعلى أية حال فإن منهج علاقات الكينونات يؤكد أن إحدى نتائج تحليل النظام هي الفهم الواضح لما تشمله الكينونات. والمنهج المبني على التطبيع يأخذ

وجهة النظر بأن تحليل النظم ينتج عنه قائمة بحقول بيانات التطبيق والعلاقات الموجودة بينها. وعند ذلك تكون عملية التطبيع مسئولة عن فصل الحقول التي تعرف كينونات عن الحقول التي تصف كينونات كما سوف نرى.

ومنهج علاقات الكينونات يستخدم رسومات علاقات الكينونات entity - relationship ويوجد مثال افتراضي له في شكل 10.17. المستطيلات الموجودة في الشكل تمثل أنواع كينونات والصناديق التي تشبه قطع الماس تمثل علاقة بين الكينونات وتمثل الدوائر خصائص. وعلى هذا يبين شكل 10.17 كينونتين A و B يوجد بينهما علاقة واحد إلى العديد X لكل منهما (لاحظ أن «1» و «M» على كل جانب من جوانب X). والكينونة A لها خاصيتين (حقلين ليصفاها) F و G. والعلاقة X لها معلومات وصفية أكثر لها H. وتمثل H معلومات عن العلاقة X أي عن أزواج من A و B غير مطبقة على حدوث A بنفسه أو حدوث B بنفسه.



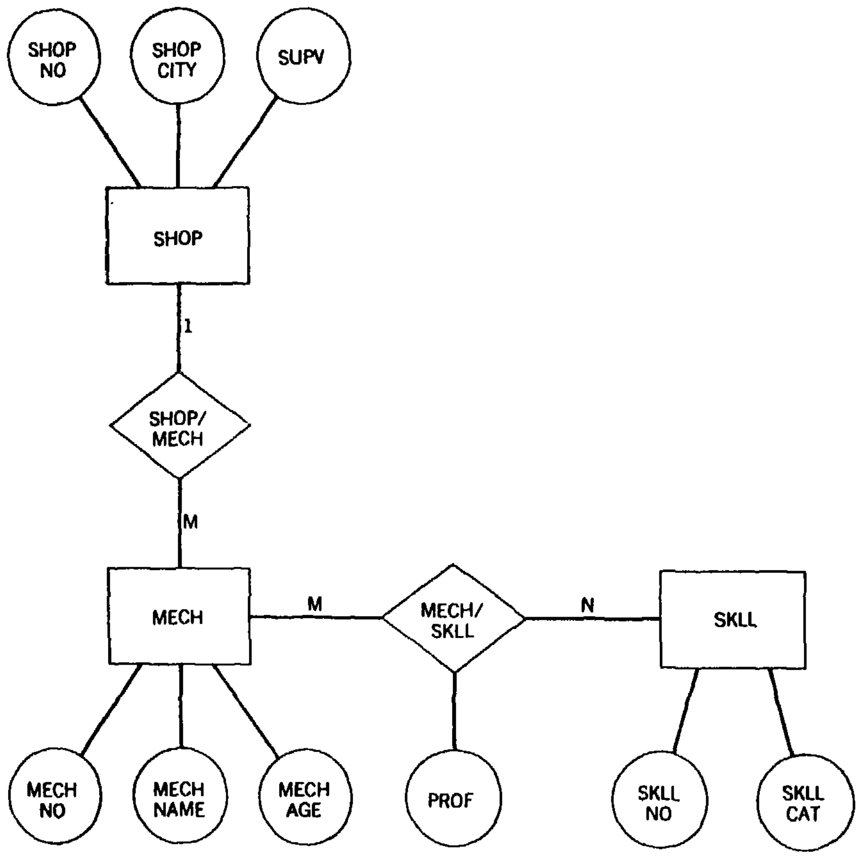
شكل 10.17
رسم الكينونة والعلاقة

ويبين شكل 10.18 رسم لعلاقات كينونات للحقول والعلاقات المبينة في شكل 10.6 و 10.7 (نفس المثال المستخدم في مناقشة تكوين عدم التطبيع). وقد تم تمييز الورش والميكانيكيين والمهارات كثلاثة كينونات في التطبيق وعلاقة من الواحد إلى العديد الموجودة بين SHOP و MECHANIC وعلاقة من العديد إلى العديد الموجودة بين MECHANIC كل منهما معروضة بوضوح. لاحظ الحقول (بما في ذلك الحقول الرئيسية) التي تصف الكينونات وبيانات PROFICIENCY التي تصف أكثر العلاقة بين MECHANICS و SKILLS.

هناك زخرفة أخرى يمكن أن تضاف إلى رسم علاقات الكينونات.

إحدى هذه الوحدات هو رمز يبين أن حدوث كينونة معينة لا يمكن أن يوجد حدوث كينونة أخرى. ووحدة أخرى هي تحديد أن إحدى الكينونات لا يكون لها مفتاح مصاحب فردي معها ويجب أن تعتمد على كينونة أخرى للتعريف المناسب.

من رسومات علاقات الكينونات إلى تكوينات شبكات CODASYL:
وأخيراً يجب أن تحول رسومات علاقات الكينونات إلى صورة مناظرة لتكوين



شكل 10.18

رسم الكينونة والعلاقة لعينة البيانات

البيانات لنظام إدارة قاعدة البيانات المستخدم. والتحويل من رسومات علاقات الكينونات عبارة عن شبكة في حد ذاته. قارن رسم علاقات الكينونات الموجود في شكل 10.18 مع شبكة CODASYL الموجودة في شكل 10.16. الرسمان متطابقان مع تمييز أن الواصل في المنتصف لعلاقة من العديد إلى العديد يجب أن يتحول إلى نوع سجل ويخدم كهدف لعلاقتين من واحد إلى العديد تحلان محل علاقة من العديد إلى العديد.

من رسومات علاقات الكينونات إلى التكوينات الهرمية: التحويل من رسم علاقات الكينونات يتطلب قليلاً من المجهود الإضافي إلا أنه ما زال أساسياً. وعلاقات من الواحد إلى العديد المبينة في رسم علاقات الكينونات تصبح ببساطة خليطاً من قطاعات عائل وتابع في هرميات DL/I وتكون علاقات من العديد إلى العديد الأساسي لعلاقات DL/I المنطقية. والصندوق الذي يشبه الماس الموجود بين كينونتين في علاقة من العديد إلى العديد يصبح تابعاً منطقياً في علاقة DL/I المنطقية وأي حقل مصاحب لهذا الصندوق يصبح بيانات متقاطعة. وعلى هذا فإن رسم علاقات الكينونات الموجود في شكل 10.8 يمكن أن يتحول إلى هرميات DL/I الموجودة في شكل 10.15.

من رسومات علاقات الكينونات إلى تكوينات علاقية وشبه علاقية: التحويل من رسم علاقات كينونات إلى مجموعة من التكوينات العلاقية أو الملفات المسطحة المعتادة لهذا الغرض هو عكس مجهود التصميم باستخدام المنهج المبني على التطبيع. مجموعة الخواص لكل كينونة في رسم علاقات الكينونات تصبح علاقة أو ملفاً مع تحول الصفة أو الخاصية الفريدة في التعريف إلى مفتاح. ولكن بالإضافة إلى ذلك فإن خاصية التعريف للكينونة من جانب واحد من جوانب علاقة من الواحد إلى العديد يجب أن تزدوج وتجعل أحد الخواص للعلاقة مبنياً على كينونة على العديد من الأوجه للعلاقة. بمجرد تعرية الرسم الشبيه بالشبكة فإن هذه هي الطريقة الوحيدة لحفظ تتبع أي حدوثات للكينونة على جانب واحد one يرتبط بحدوث على عدة many جوانب. فإذا كانت هناك هرمية كينونات متعددة المستويات فيجب أن ينتشر مثل هذا الإزدواج خلال الهرمية عند تحويلها

إلى علاقات أو ملفات. بالإضافة إلى ذلك يجب إنتاج علاقات أو ملفات من علاقات من العديد إلى العديد التي توجد في رسم علاقات الكينونات. ومفتاح مثل هذه العلاقة يحتوي على التعرف أو الحقول الرئيسية للكينونتين المشمولتين. والحقول غير الرئيسية سوف تصبح معدلات للعلاقة (الحقول المصاحبة للصندوق الذي يشبه قطه الماس).

اعتبر تحويلاً لرسم علاقات كينونات الموجود في شكل 10.18، أي علاقات الصورة الطبيعية الثالثة الموجودة في شكل 10.12. لاحظ أن كينونات الورشة والميكانيكي والمهارة يكون كل منها أساساً للعلاقة. علاقة Mechanic يجب أن تحتوي على نسخة من مفتاح علاقة الورشة (SHOP NO) لأنه يكون هدف في علاقة من واحد إلى العديد. وعلاقة Proficiency في شكل 10.12 تستخلص من علاقة من العديد إلى العديد الموجودة بين MECHANIC و SKILL في شكل 10.18. لاحظ أن المفتاح MECH NO - SKILL NO مكون من حقول تعريف لكينونتي الميكانيكي والمهارة وأن الحقل غير الرئيسي PROFICIENCY قد قام بتشكيل الحقل المتصل بواصل العلاقة ذات الشكل الماسي.

وشكل 10.19 هو خريطة تلخيصية للأربعة مناهج لنظم إدارة قواعد البيانات وأسلوبي تصميم قواعد البيانات الذي نوقشا. ومن المهم ملاحظة أن تطبيع البيانات والمناهج المبنية على علاقات الكينونات مع بعضها لتصميم قاعدة البيانات تستخدم في بعض الأحيان ترادفياً في تشييد قاعدة بيانات واحدة. مثال ذلك، يمكن أن يضع المصمم افتراضات ابتدائية عن الكينونات والخواص والعلاقات في نمط منهج الكينونة والعلاقة ثم يختبر هذه الافتراضات مستخدماً تطبيع البيانات.

تصميم قاعدة البيانات يستخدم في بعض الأحيان في ترادف تشييد قاعدة بيانات فردية. مثال ذلك قد يضع المصمم افتراضات أولية عن الكينونات والخواص والعلاقات في صورة منهج علاقات الكينونات ثم يختبر هذه الافتراضات باستخدام تطبيع البيانات.

	Normalization Approach	Entity-Relationship Approach
Relational	Normalization	Entity-relationship diagram "dismantled" into a set of relations
Pseudo-Relational	Normalization plus join tables or indices	Entity-relationship diagram "dismantled" into a set of relations plus join tables or indices
Hierarchical	Normalization followed by hierarchical structuring	Entity-relationship diagram converted to a hierarchical structure
Network	Normalization followed by network structuring	Entity-relationship diagram converted to a network structure

شكل 10.19

خريطة تلخيص تصميم قاعدة البيانات (كل النتائج قابلة للتعديل لأسباب خاصة بالأداء)

وسائل أخرى لتصميم قاعدة البيانات :

Other Tools Database Design

هناك أنواع عديدة من وسائل نظم البرامج التي يمكن أن تساعد في عملية تصميم قاعدة البيانات.

وسائل تلقائية لتصميم قاعدة البيانات :

خلال السنوات الأخيرة القليلة ظهر العديد من منتجات نظم البرامج الخاصة بالتصميم لقواعد البيانات. وتقليدياً فإن هذه المنتجات تتطلب قائمة من الحقول تتكون منها قاعدة البيانات والعلاقات الموجودة بين هذه الحقول مع بعضها البعض. وبصفة عامة فهي قادرة على تنفيذ تصميم منطقي لقاعدة البيانات. وفي بعض الحالات مع تدخل جزئي من العاملين يحدث تصميم واقعي لنظام إدارة قاعدة بيانات معين. وكجزء من عملها فإنها يمكنها إجراء تنقيح على بيانات المدخلات وفي عمل ذلك يمكنها كذلك استكشاف الأخطاء وعدم التماسك.

وقد لاقت برامج تصميم قواعد البيانات نجاحاً مختلطاً في هذا المجال. وأكبر اعتراض من المستفيدين هو أن تحديد الحقول والعلاقات بالعمق والدقة

اللازمين (لأي برنامج كمبيوتر) يستغرق وقتاً طويلاً جداً. وهذا على أية حال هو اتجاه سيء حيث أن مجموعة الحقول والعلاقات يجب أن تحدد بدقة لتصميم قاعدة البيانات سواء استخدمت وسيلة تلقائية أم لا. ونتيجة عمل غير ذلك هي الحصول على تصميم ضعيف لقاعدة البيانات.

وسائل التنبؤ بالأداء وتوجيهه :

بعد إتمام تصميم قاعدة البيانات المنطقية والواقعية فيكون هناك طرق مختلفة متاحة للتنبؤ بأداء النظام المتوقع. ولكي نكون دفيقين أكثر بالنسبة لأي استخدام واقعي فيجب أن تأخذ هذه الطرق في اعتبارها معلومات هائلة تقع خارج هيكل قاعدة البيانات. مثال ذلك يجب أن تعتبر التكرار النسبي للحدوث لأنواع السجلات المختلفة وأنماط استدعاء برامج التطبيقات والمنافسة على الموارد من التطبيقات الأخرى التي تستخدم نفس قواعد البيانات أو غيرها. وبعض هذه الطرق أجرى عليه عمليات تجعله تلقائياً جزئياً مثل مقاطع المحاكاة.

خاتمة : Conclusion

بعد إتمام عملية تصميم قاعدة البيانات يتحول تركيز عملية تطوير التطبيق إلى المبرمجين. ويفترض أن تصميم البرنامج يحدث في نفس الوقت مع تصميم قاعدة البيانات. وعلى هذا يمكن أن يتواجد المبرمجون مرة واحدة مع شفرتهم المحددة لهم مع تكوينات البيانات اللازمة لاستخدامها.

وبعد إعداد الشفرة توجد خطوات في التسلسل. وتشمل هذه مستويات مختلفة من اختبارات البرامج والتشييد والصيانة أثناء التشغيل. وكل الخطوات من البرمجة وخلال الصيانة تتمتع بشدة العمل المستمر والدقيق الذي يكون قد حدث في المراحل السابقة من عملية تطوير التطبيق خلال تصميم قاعدة البيانات.

References

مراجع:

Chen. P.P., «Applications of the Entity - Relationship Model», Procedings of the NYU Symposium on Database Design, May 18 - 19, 1978, pp. 25 - 33.

Chen, P.P., and Yao, S.B., «Design and Performance Tools for Data Base Systems», Proceedings of the Third International Conference on Very Large Data Bases, October 6 - 8, 1977, pp. 3 - 15.

Curtice, R.M., «Data Base Design Using a CODASYL System», Proceedings of the ACM Annual Conference, November, 1974, pp. 473 - 480.

Curtice, R.M., and Jones, P.E., Logical Data Base Design, Van Nostrand Reinhold, New York, 1982.

Data, C.J., An Introduction to Database Systems, 3rd ed., vol. 1, Addison - Wesley, Reading, MA, 1981.

DeMarco, T., Structured Analysis and System Specification, Yourdon, New York, 1978.

Gane, G., and Sarson, T., Strutured Systems Analysis: Tools and Techniques, Prentice - Hall, Englewood Cliffs, NJ, 1979.

Gerritsen, R., «A Preliminary System for the Design of DBTG Data Structures», Communications of the ACM, vol. 18, no. 10, October, 1975, pp. 551 - 557.

Gillenson, M. L., and Goldberg, R. Strategic Planning, Systems Analysis, and Database Design, Wiley, New York, 1984.

Hawryszkiewycsz, I. T., Database Analysis and Design, Science Research Associates, Chicago, 1984.

Hubbard, G. U., Computer - Assisted Data Base Design, Van Nostrand Reinhold, 1981.

IBM Corporation, «Business Systems Planning - Information Systems Planning Guide», 3rd ed., IBM form No. GE20 - 0527 - 3, 1981.

Inmon, W. H., Effective Data Base Design, Prentice - Hall, Englewood Cliffs, NJ, 1981.

Navathe, S.B., and Schkolnick, M., «View Representation in Logical Database Design, «Proceedings of the SIGMOD International Conference on Management of Data, May 31, June 1 - 2; 1978, pp. 144 - 156.

Ng, P.A., «Further Analysis of the Entity - Relationship Approach to Database Design», IEEE Transactions on Software Engineering, vol. SE - 7, no. 1, January, 1981, pp. 85 - 99.

Novak, D.O., and Fry, J.P., «The State of the Art of Logical Database Design», Proceedings of the Fifth Texas Conference on Computing Systems, October 18 - 19, 1976, pp. 30 - 38.

Perkinson, R. C., Data Analysis: The Key to Data Design, QED Information Sciences, Wellesley, MA, 1984.

Perron, R., Design Guide for CODASYL Data Base Management Systems, QED Information Sciences, Wellesley, MA, 1981.

Teorey, T. J., and Fry, J. P., «The Logical Record Access Approach to Database Design», Computing Surveys, vol. 12, no. 2, June, 1980, pp. 179 - 211.

Teorey, T. J., and Fry, J. P., Design of Database Structures, Prentice - Hall, Englewood Cliffs, NJ, 1982.

Vetter, M., and Maddison, R. N., Database Design Methodology, Prentice/Hall International, Englewood Cliffs, NJ, 1981.

Weinberg, V., Structured Analysis, Yourdon Press, New York, 1979.

أسئلة وتمارين: Questions and Exercises

١ ـ ما تصميم قاعدة البيانات؟

٢ ـ لماذا يجب أن نهتم بالتخطيط الإستراتيجي وتحليل النظم لنظم المعلومات وذلك بالنسبة إلى تصميم قواعد البيانات؟

٣ ـ ما التخطيط الاستراتيجي لنظم المعلومات؟ ما أهدفه؟

٤ ـ وضح كيف يمكن أن يساعد تحليل عمليات الأعمال وفئات البيانات في تحقيق أهداف التخطيط الاستراتيجي لنظم المعلومات.

٥ ـ ما تحليل النظم؟ وأين يتناسب وجوده داخل دورة تطوير النظام؟

٦ ـ صف منهج تحليل النظم المرتب. هل تعتقد أنه يمكن أن يعمل جيداً في مواقف تطبيقات عملية؟

٧ ـ ما شعورك بالنسبة إلى مفاهيم تحليل النظم المرتب في حفظ رسومات تحليل النظم خالية من المكونات الطبيعية للكمبيوتر والإشارة إليها وفي صورة تجعل المستفيد غير المتخصص يتفهمها؟

٨ ـ صف السريان الذي يمكن اعتباره للبيانات في التخطيط الاستراتيجي لنظم المعلومات في تحليل النظم وفي تصميم قاعدة البيانات.

٩ ـ ناقش المواضيع التي تجعل من عمل تصميم دقيق لقاعدة البيانات ضرورياً.

١٠ ـبكلمات من عندك صف عملية تطبيع البيانات.

١١ ـناقش عملية تطبيع البيانات بالنسبة إلى قواعد البيانات العلاقية. كيف يرتبط المنتج النهائي لتطبيع البيانات بتصميم قواعد البيانات العلاقية؟ ماذا عن موقع حقول الوصل؟

١٢ ـصف عملية التحول من الجداول التي أجري لها تطبيع إلى تصميمات قواعد بيانات هرمية وشبكية.

١٣ ـما الفرق بين تصميم قاعدة البيانات المنطقية وتصميم قاعدة البيانات الواقعية؟ وضح إجابتك.

١٤ ـصف منهج علاقات الكينونات لتصميم قواعد البيانات.

١٥ ـ قارن المنهج المبني على تطبيع العلاقات لتصميم قاعدة انبيانات بالمنهج المبني على علاقات الكينونات. ما مميزات وعيوب كل منهما؟

١٦ ـتطور شركة Wign - N - Prayer Airlines نظام تاريخي لصيانة الطائرات للمساعدة في مقابلة التعليمات الحكومية للسلامة. وبالإضافة الى إنتاج معلومات عن الطائرات الفردية فيتوقع للنظام أن يقدم استجابة سريعة كذلك للاستفسارات عن الطائرات الفردية ومواقع صيانتها. صمم:

أ ـ قاعدة بيانات علاقية.

ب ـ قاعدة بيانات هرمية.

جـ ـ قاعدة بيانات شبكية.

د ـ قاعدة بيانات شبه علاقية.

لهذا التطبيق وفيما يلي الحقول وبعض الاعتمادات والقيود لاحظ أن بعض هذه الحقول يمكن أن يشمل معلومات يمكن تمثيلها بطرق غير الحقول. ضع كذلك أي افتراضات إضافية تراها عن البيانات في الأماكن التي تشعر أن وصف المشكلة مبهم فيها.

الحقول:

أ ـ رقم مسلسل للطائرة.
ب ـ نوع الطائرة.
جـ ـ سعة ركاب الطائرة.
د ـ موقع الصيانة.
هـ ـ مدير الصيانة.
و ـ رقم مبنى الصيانة.
ر ـ اجراءات الصيانة.
ز ـ طول وقت الصيانة.
س ـ تكرارات العملية.
ش ـ تاريخ تنفيذ العملية على طائرة معينة.
ع ـ تاريخ الطائرة التالية المجدولة للعملية.
غ ـ الموقع القادر على إجراء العلمية.
م ـ اسم الآلة.
ن ـ الميكانيكي المؤهل لتنفيذ العملية.
ص ـ تعليم الميكانيكي.
ض ـ المعدات المحددة للميكانيكي (برقم التعريف).

الاعتمادات والقيود:

أ ـ الموقع المعطى يمكن أن ينفذ العديد من العمليات.

ب ـ العملية المعطاة يمكن أن تنفذ في موقع واحد فقط (لكل الطائرات).

جـ ـ الموقع يمكن أن يكون به العديد من المباني.

د ـ يجب أن تحفظ قاعدة البيانات سجلات تاريخية عن كل تنفيذات عمليات الصيانة وذلك طوال فترة عمل الطائرة.

هـ ـ يمكن أن يكون الميكانيكي مؤهلاً لتنفيذ عمليات عديدة ولكنه قد لا ينفذ إلا العمليات المتاحة في الموقع الذي يعمل فيه فقط.

و ـ قد يكون لدى الميكانيكي وحدات تعليم متعددة وكذلك معدات متعددة محددة له.

١٧ ـلقد قررت الإدارة المكتبية الحكومية Fedral Bureaucracy Administration (FBA) تنفيذ نظام معلومات جديد يصمم لحفظ بيانات عن الولايات والمدن وبعض المسئولين المختارين. صمم.

أ ـ قاعدة بيانات علاقية.

ب ـ قاعدة بيانات هرمية.

جـ ـ قاعدة بيانات شبكية.

د ـ قاعدة بيانات شبه علاقية.

لهذا التطبيق. الحقول والاعتمادات والقيود تلي ذلك لاحظ أن بعض الحقول يمكن أن تحدد معلومات يمكن أن تمثل بطرق أخرى غير الحقول. ضع كذلك أي افتراضات إضافية عن البيانات في المواقع التي تشعر فيها أن المشكلة مبهمة.

الحقول:

أ ـ المدينة.

ب ـ اسم عضو مجلس المدينة.

جـ ـ وظيفة عضو مجلس المدينة.

د ـ تعداد المدينة.

هـ ـ اسم المحافظ.

و ـ رقم الضمان الاجتماعي للمحافظ.

ر ـ لجنة السيناتور.

ز ـ موقع لجنة السيناتور.

س ـ عضو لجنة السيناتور.

ش ـ سنة تشكيل لجنة السيناتور.

ع ـ اسم السيناتور.

غ ـ رقم الضمان الاجتماعي للسيناتور.

م ـ عضوية السيناتور في اللجنة.

ن ـ رتبة السيناتور في اللجنة.

ص ـ الولاية.

ض ـ تعداد الولاية.

الاعتمادات والقيود:

أ ـ أفراد السيناتور حكوميين بدلاً من كونهم من الولاية.

ب ـ يتم الاتصال بالبيانات عن طريق الولاية بصفة عامة لكنه قد يحدث عن طريق المدينة أو السيناتور بصفة مؤقتة.

جـ ـ يخدم السيناتور في العديد من اللجان في نفس الوقت.

د ـ أعضاء لجنة السيناتور ليسو سيناتور ولكنهم قد يخدمون كلجنة سيناتور.

هـ ـ كل مدينة لها أعضاء مجلس مدينة عديدين.

و ـ كل عضو مجلس مدينة لديه عمل واحد وقتاً كاملاً.

ر ـ اسماء المحافظين ليست فردية.

ز ـ اسماء المدن فردية داخل الولايات فقط.

١٨ ـيريد اتحاد لاعبي كرة القدم Football Players Association تصميم قاعدة بيانات لدعم نظام معلومات يحفظ بيانات تاريخية عن اللاعبين والفرق. وتشمل الاستفسارات المعتادة الحصول على كل أو جزء من المعلومات المرتبطة بالفريق عن لاعب معين أو كل أو جزء من المعلومات المرتبطة باللاعبين عن فريق معين. كما يكون هناك استفسارات موجهة كذلك للملكية. صمم.

أ ـ قاعدة بيانات علاقية.

ب ـ قاعدة بيانات شبكية.

جـ ـ قاعدة بيانات هرمية.

د ـ قاعدة بيانات شبه علاقية.

لهذا التطبيق. الحقول والاعتمادات والقيود تلي ذلك لاحظ أن بعض الحقول يمكن أن تحدد معلومات يمكن أن تمثل بطرق أخرى غير الحقول. ضع كذلك أي افتراضات إضافية عن البيانات في المواقع التي تشعر فيها أن المشكلة مبهمة.

الحقول:

أ ـ اسم الفريق.

ب ـ رقم تعريف اللاعب.

جـ ـ اسم اللاعب.

د ـ اسم مالك الفريق.

هـ ـ الفريق الحالي للاعب.

و ـ الفريق السابق للاعب.

ر ـ عدد سنوات لعب اللاعب في الفريق.

ز ـ متوسط الراتب السنوي للاعب في الفريق (لاعب في فريق واحد طوال سنوات لعبه لهذا الفريق).

س ـ موقع اللاعب.

ش ـ مدينة سكن اللاعب.

ع ـ ولاية سكن اللاعب.

غ ـ مدينة الفريق.

م ـ ولاية الفريق.

ن ـ حظ الفريق.

ص ـ عدد سنوات ملكية المالك للفريق.

ض ـ المدرب الرئيسي.

ك ـ ترتيب جودة اللاعب السنوي (بالنسبة لكل الفرق التي لعبت هذا العام).

ل ـ تواريخ الملكية.

الاعتمادات والقيود:

أ ـ يجب أن تحفظ البيانات طوال فترة حياة كل لاعب.

ب ـ يمكن أن يكون اللاعب معيناً بصورة وقتية فقط.

ج ـ يجب أن تحفظ البيانات عن كل مالك سابق للفريق وعن المالك الحالي كذلك.

د ـ لكل فريق مالك واحد فقط في نفس الوقت.

الفصل الحادى عشر
قواميس البيانات

مقدمة : Introduction

مثل أي جزء من أجزاء الأعمال تحتوي مؤسسة تشغيل البيانات على عدد من المكونات الشاملة التي يجب أن تدار. فلديها مخزون من البيانات ولديها مدخلات وهي البيانات الجديدة والطلبات الخاصة بالبيانات. ولديها منتجات وهي التقارير والاستجابات على الطلبات الخاصة بالبيانات وما إلى ذلك. ولديها معدات وهي أجهزة الكمبيوتر وكل الوحدات المصاحبة لها. ولديها مجموعة من الأفراد الذين يؤدون العمل. وفي الأيام الأولى لتشغيل البيانات بما في ذلك في وقتنا الحالي في مؤسسات تشغيل البيانات الصغيرة يمكن لمدير لامع لتشغيل البيانات بأخذه بملاحظات أولية مكتوبة هنا وهناك أن يتذكر ما يكفي من التفاصيل عن كل مكونات العمل لإدارته بكفاءة. وفي الواقع فإن حقيقة الأمر هي أن مدير تشغيل البيانات التقليدي في الماضي لم يفكر كثيراً في الإدارة بمفهوم الكتاب الجامعي. وقد مال مدير تشغيل البيانات إلى أن يعمل تحت ضغط من جميع الجهات التي تعمل بالإدارة مما جعل هناك تمديدات لأوقات انتهاء من الأعمال تليها أوقات أخرى لأعمال محددة دون التفكير في التأثيرات الأعمق.

وحالياً نجد أن أقسام تشغيل البيانات عبارة عن أعمال ضخمة تشمل آلاف من العاملين في الشركات الكبيرة ومئات في الشركات الأخرى. وقد ازداد عدد مديري تشغيل البيانات الذين يحملون لقب نائب الرئيس والعديد منهم لم يأتوا من خلال مراتب تشغيل البيانات وإنما أتوا من مواقع أخرى. وقد تميزوا بأنهم قاموا بإدارة عمل تشغيل البيانات كأحد مجال الأعمال داخل مجالات الأعمال الأخرى. وقد أصبحت الضغوط هائلة مع ازدياد اعتماد أعمال الشركات أكثر وأكثر على

تشغيل البيانات في أداء وظائفها اليومية. وفي نفس الوقت فإن مديري تشغيل البيانات أصبحوا مسؤولين عن أمن البيانات ومراقبة التكاليف وتغيير بقية الشركة بالنسبة للبيانات المتاحة لهم وما إلى ذلك. هل هناك أي مساعدة لمدير تشغيل البيانات في إدارة بيئة تشغيل البيانات؟ تبدأ الإجابة بنظرة إلى أنواع المساعدة المتاحة لهم من الرؤساء الوظيفيين في المؤسسة.

تذكر أن الغرض من تشغيل البيانات هو مساعدة كل المناطق الوظيفية للشركة في أدائها بكفاءة أكبر وفعالية أكثر. فالمدير الحالي يحصل على دعم هائل من المحاسبة وأعداد النماذج من استخدامات الكمبيوتر الأولى في الشركة. ومدير الأفراد لديه كل شيء من عناوين المنازل إلى المهارات المكتسبة عن كل موظف من الموظفين ورئيس التصنيع لديه مخازن تجدد بصفة دائمة للمواد الخام. وتشمل القوائم كل شيء يمكن أن يتواجد في الشركة.

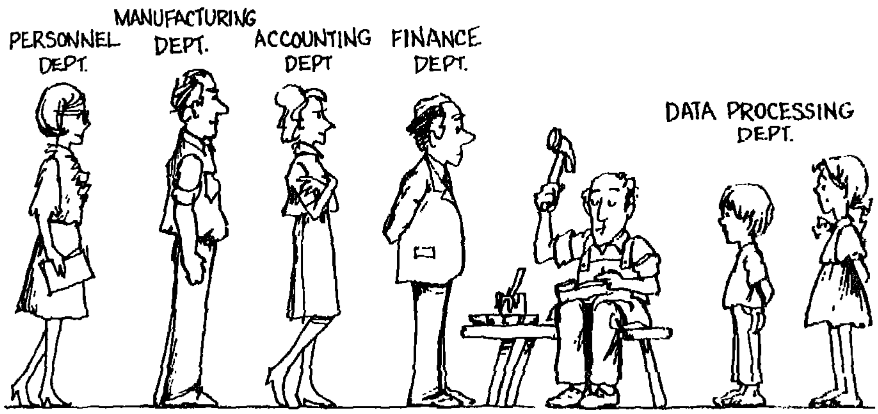
لكن ماذا عن المساعدة لمدير تشغيل البيانات؟ تاريخياً كان مديروا تشغيل البيانات مشغولين دائماً في تطبيقات تشغيل البيانات لمقابلة احتياجات الآخرين مع إهمالهم لاحتياجاتهم الخاصة. وهذا يذكرنا بأطفال صانع الأحذية الذين يحصلون على أحذيتهم آخر الناس أي بعد حصول كل العملاء على أحذيتهم كما هو موضع في شكل 11.1. وقد بدء حديثاً فقط اهتمام العاملين في تشغيل البيانات اهتماماً جاداً في التطبيقات الخاصة بهم. ومن هذا المجهود ظهرت مجموعة من ملفات الكمبيوتر التي يستخدمها قسم تشغيل البيانات في المساعدة في إدارة أنشطته الخاصة. وقد سميت مجموعة الملفات هذه بقواميس البيانات data dictionary.

تعريف وخواص: Definition and Characteristics

دعنا نستكشف قواميس البيانات في خطوتين: في هذا القسم نلقي نظرة على مفهوم قواميس البيانات على مستوى شامل. وفي الأقسام التالية نتعرض لها بشيء من التفاصيل.

يبين شكل 11.2 ملفين أكثر شيوعاً في مؤسسات تشغيل البيانات لإحدى شركات تصنيع تقليدية. فقسم الأفراد يهتم بحفظ تتبع العاملين بالشركة والذي نشير إليه بأنه فئة كينونة (كل شخص في الكينونة). أنواع الحقيقة بأنهم يخزنون

بيانات لكل عامل تشمل رقم تعريفه واسمه واسم عائلته وما إلى ذلك (كل حقيقة هي خاصية لهذه الكينونة). وبالمثل فإن أحدى فئات الكينونة التي تهم قسم التصنيع هي فئة بالعناصر الموجودة في مخازن التصنيع ولكل عنصر يجب أن يخزن رقم الجزء واسمه وما إلى ذلك.



شكل 11.1

أطفال صانع الأحذية يحصلون على أحذيتهم آخر الناس

وعلى هذا فقسم التصنيع وقسم الأفراد لديهم ملفات كمبيوتر (أحد ملفات كل قسم موضح في شكل 11.2)، تساعدهم في إدارة أعمالهم. ما هو الدعم المكافيء الذي يجب أن يحصل عليه قسم تشغيل البيانات؟

يبين شكل 11.3 ملفين لما يجب أن يكون عليه 5 أو 10 ملفات قواميس بيانات data dictionary وهي مجموعة من الملفات مصممة لمساعدة قسم تشغيل البيانات في إدارة أعماله. فإذا أراد الأفراد تتبع احتياجات الأشخاص والتصنيع لمعرفة عناصر مخزونهم وذلك ضمن أشياء أخرى، فماذا يريد أن يتتبعه تشغيل البيانات؟ هناك شيئان لكينونات يجب أن يديرها تشغيل البيانات وهما الحقول والملفات. وعلى هذا يبين شكل 11.3 ملف حقول field file وملف ملفات files file.

EMPNO	EMPLAST	EMPFIRST	...
06337	Jones	Fred	...
09155	Smith	Susan	...
16840	Adams	John	...
. . .			

Personnel File

PARTNO	PARTNM	...
115248	Nut	...
124770	Bolt	...
173771	Screw	...
. . .		

Inventory File

شكل 11.2

ملفان تشغيل بيانات تقليديان

FIELDNM	FIELDTYP	FIELDLEN	...
EMPNO	Num	5	...
EMPLAST	Alpha	20	...
EMPFIRST	Alpha	10	...
PARTNO	Num	6	...
PARTNM	Alpha	20	...
. . .			

Fields File

FILENM	FILELEN	...
Personnel	1000	...
Inventory	15000	...
.		
.		
.		

Files File

شكل 11.3

جزء من قاموس البيانات لملفات الشكل 11.2

فإذا قرر قسم الأفراد أن رقم العامل واسم العائلة واسم العامل وما إلى ذلك هي خواص يجب تخزينها لكل عامل وقرر قسم التصنيع أن رقم الجزء واسم الجزء وما إلى ذلك فما هي الخواص لحقول الملفات التي يشعر تشغيل البيانات بأنها يجب أن تخزن لكل منهما في قاموس البيانات؟ يبين شكل 11.3 أنه بالنسبة للحقول فـإن بعض الخـواص هي أسماء حقـول (FIELDNM) وأنـواع حقـول (FIELDTYP) وأطـوال حقـول (FIELDLEN) وللملفـات أسماء ملفـات (FILEVM) وأطوال ملفات (FILELEN).

في ملف الأفـراد يمثل كـل صف أحد الكينـونات : أحـد العاملين، وفي ملف المخزون يمثل كل صف أحد الأجـزاء. والمكافيء في قـاموس البيـانات هـو أن كل صف في ملف الحقـول field file يمثـل أحـد الحقـول في أحـد ملفـات بيـانـات التطبيقات التي يقوم بتشغيلها قسم تشغيـل البيانـات وكل صف في ملف الملفـات files file يمثل أحد ملفات التطبيقات التي يقوم بتشغيلها قسم تشغيل البيـانات. وعـلى هذا ففي هـذا المثـال كـل صف في ملف الحقـول في شكـل 11.3 يمثـل أحـد حقول أحد ملفات شكل 11.2. كما أن كل صف في ملف الملفـات في شكل 11.3 يمثل أحد ملفات شكل 11.2.

ما سبق وصفه حتى الآن، تخزين بيانات عن كينونات تشغيل بيـانات، هـو أحـد بعدين رئيسيـين في قامـوس البيانـات أما البعـد الرئيسي الثـاني فهـو تخـزين معلومات عن العلاقـات relationships بين الكينـونات. وليس كـافياً حفظ تتبـع عناصر هامة في بيئة تشغيل البيانات بنفسها فمن المهم كذلك حفظ معلومـات عن كيفية ارتباط هذه العناصر بعضها بالبعض الآخر. والتوصيلات غير الـرسمية بـين الملفـين المـوجـودين في شكـل 11.4 (وهـو نفس شكـل 11.3 مـع إضـافـة هـذه التوصيلات) تـوضح العـلاقات بـين الملفات والحقـول المبينة والحقـائق أن الحقول EMPNO و EMPLAST و EMPFIRST هـي في مـلف الأفـراد Personnal

FIELDNM	FIELDTYP	FIELDLEN	...
EMPNO	Num	5	...
EMPLAST	Alpha	20	...
EMPFIRST	Alpha	10	...
PARTNO	Num	6	...
PARTNM	Alpha	20	...
⋮			

Fields File

FILENM	FILELEN	...
Personnel	1000	...
Inventory	15000	...
.		
.		
.		

Files File

شكل 11.4

جزء من قاموس البيانات يبين العلاقات

والحقول PARTNO و PARTNM هي في ملف المخزون Inventory هامة جداً ليعرفها مدير تشغيل البيانات. فعلى سبيل المثال يمكن أن يكون نفس الحقل مستخدماً في ملفات عديدة ويجب أن يكون هناك شيء من التنسيق لهذا الاستخدام المتعدد.

ونظراً لأن قاموس البيانات يحتوي على معلومات وصفية عن ملفات الأفراد وملفات التصنيع وما إلى ذلك فيجب أن نشير إلى قاموس البيانات على أنه تخزين «لبيانات عن بيانات» أو «ما وراء البيانات metadata».

محتويات القاموس: Dictionary Contents

دعنا نلقي نظرة أكثر تفصيلاً على أنواع الكينونات والخواص والعلاقات التي يمكن أن توجد في قاموس البيانات.

الكينونات:

هناك عدد من كينونات بيئة تشغيل البيانات المتفق عليها بصفة عامة لتتواجد في قاموس البيانات. وطبقاً لاحتياجات المؤسسة يمكن أن تكفي فئة جزئية منها أو يكون مطلوباً كينونات إضافية.

ونوع الكينونات الأكثر وضوحاً لاعتباره في قاموس البيانات هو النوع المستخدم في المثال السابق: البيانات المرتبطة بصورة مباشرة بالكينونة. وكما سبق أن رأينا فإن الحقول والملفات تمثل هذه الفئة. ماذا عن السجلات ككينونات؟ يمكن القول أن ملف السجلات record file يجب أن يحتوي على مثل هذه المعلومات كطول السجل ومجموعة من المشيرات لملف السجلات records file محدداً أي أنواع السجلات الموجودة في الملف. ولكن إذا كان هناك نوع سجلات واحد في الملف وهذه تمثل الحالة المعتادة (للملفات الخطية) فعند ذلك يمكن دمج كل هذه المعلومات ـ طول السجل وطول الملف والمشيرات إلى الحقول الموجودة في سجلات الملف في ملف ملفات files file. فإذا كانت الشركة تستخدم نظام إدارة قاعدة بيانات يستخدم تكوينات هرمية أو تكوينات شبكية أو تكوينات ملفات متصلة أو أي تكوينات خاصة أخرى فعند ذلك يكون من المناسب وجود كينونات لها موضوعه في قاموس البيانات كذلك.

هناك فئة أخرى لكينونات بيئة تشغيل البيانات وهي مجموعة من مكونات نظم البرامج الموجودة في تشغيل كينونات البيانات التي سبق تسميتها. ونوع الكينونات الأكثر وضوحاً في هذه الحالة هو البرنامج. وطبقاً لنظم تشغيل البيانات المستخدمة توجد أنواع أخرى تشمل ما يلي: نظم التشغيل - الأعمال - نظم إدارة قاعدة البيانات وما إلى ذلك. وهناك نوع كينونات قريب من هذا النوع وهو فئة مجموعات التحكم اللازمة لنظام إدارة قاعدة البيانات أو أي نظم أخرى تصل البيانات وتشغيلها. ومن المهم تتبع مسار نظم البرامج التي تجري تشغيلاً على البيانات التي يتم تشغيل البيانات عليها. ويمكن أن يشمل ذلك نظم الكمبيوتر المختلفة الموجودة في المنظمة ومجموعات الأقراص التي يوجد الملف عليها.

وهناك نوع كينونة لبيئة تشغيل البيانات شائع الاستخدام وهو الأفراد. من الواضح من وجهة نظر المسئولية وكذلك من وجهة نظر الأمن أنه من الضروري حفظ معلومات عن الأفراد العاملين في تشغيل البيانات. كما أن وسائل المخرجات المختلفة مثل التقارير يجب أن تكون كينونات للقاموس. بالإضافة إلى ذلك توجد أنواع كينونات بيانات شائعة الاستخدام للقاموس. وقد انتجت التجارب مع الاعتياد على قواميس البيانات أنواع كينونات أخرى للقاموس واسعة المدى. وتشمل هذه الكينونات إدخال اتصالات البيانات والتخطيط الاستراتيجي لنظم المعلومات (مثل طريقة BSP من شركة IBM).

الخواص:

هناك فئتان للخواص بالنسبة لكينونات قاموس البيانات: الخواص التي لها طبيعة عامة ويمكن أن توجد مع أي من الكينونات. والخواص الخاصة بكينونات محددة من القاموس.

والشيء الأكثر وضوحاً للخواص والعامة اسمها. ومن الواضح أن أي كينونة في القاموس يجب أن يكون لها اسم أو أي شيء آخر يعرفها. وعادة ما تسمح القواميس بمواصفات بأسماء مستعارة لأسماء الكينونات. بالإضافة إلى ذلك تقبل الكثير من القواميس وصفاً نصياً حراً للكينونة.

وحتى الآن فمعظم خواص القاموس انتشاراً هي الخواص الخاصة بكينونات محددة. وتشمل هذه الفئة ما يلي:

مدى قيمة الحقل وطول السجل وعنوان منزل أحد الأشخاص وسعة وحدة التخزين من نظم المكونات واللغة التي يكتب بها البرنامج وما إلى ذلك.

لعلاقات:

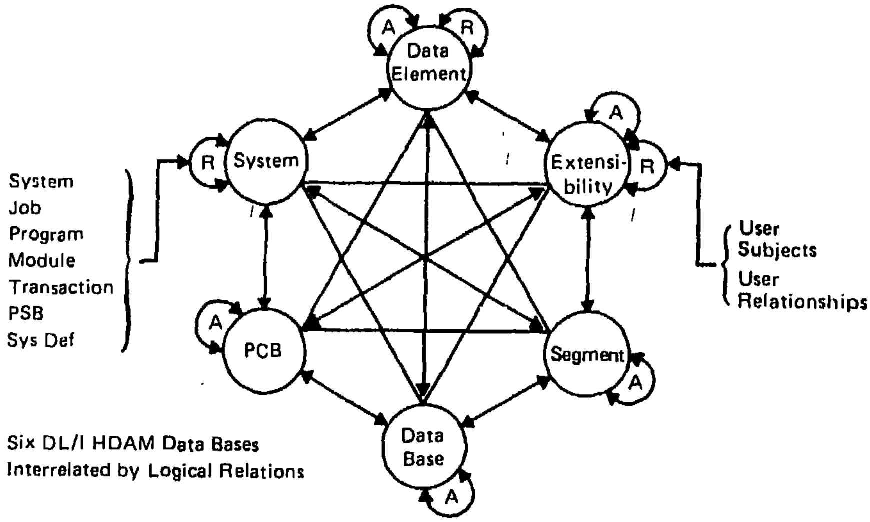
العلاقة بين أي زوج من كينونات بيانات القاموس يمكن أن يكون لها قيمة بالنسبة الى بعض أجزاء منظمة تشغيل البيانات. فبعض قواميس البيانات تحدد مجموعة محددة من العلاقات الموجودة بين كينوناتها التي صممت لمعالجتها. والبعض الآخر يسمح بالعلاقة بين كل زوج من الكينونات المستخدمة.

وفيما يلي بعض العلاقات الشائعة في القاموس والتي تشمل الكينونات المرتبطة مع بعضها البعض:

* تكوين السجل: أي الحقول تظهر في أي السجلات.
* الأمن: أي الأفراد يسمح لهم بالاتصال بأي الملفات.
* تأثير التغيير: أي البرامج يمكن أن تتأثر بالتغييرات التي تحدث في أي الملفات.
* المقاومة الطبيعية: أي الملفات موجودة على أي مجموعة أقراص.
* متطلبات البرنامج من البيانات: أي البرامج تستخدم أي الملفات.
* المسؤولية: أي الأفراد مسؤول عن تجديد أي الملفات.

وكمثال يبين شكل 11.5 الكينونات والعلاقات الموجودة في قاموس بيانات DB/DC من شركة IBM. وهناك أربعة من أنواع الكينونات لها قواعد بيانات DL/I مخصصة لها: قاعدة بيانات (DL/I هرميات أو فئات بيانات ليست DL/I) وقطاعات (تشمل سجلات غير DL/I) وعناصر (حقول) و PCB (مجموعة تحكم DL/I). وهناك مجموعة أخرى تشترك في قاعدة DL/I كذلك وتشمل المجموعة أنواع كينونات مثل البرنامج والنظام والعمل والعملية التحويلية والأفراد من البرنامج و PSB (نوع آخر من مجموعات تحكم DL/I). وهناك قاعدة بيانات

سادسة تسمى بقاعدة بيانات قابلة للتوسع. وباستخدام قاعدة البيانات هذه يستطيع المستفيد أن يعرف أنواع كينونات جديدة وخواص لهذه الأنواع ويمكنه انتاج علاقات بين أنواع الكينونات المعرفة حديثاً هذه وأي من أنواع الكينونات الأخرى التي يعرفها المستفيد. وتوضح الخطوط الموجودة في شكل 11.5 العلاقات.



شكل 11.5

الكينونات والعلاقات من قاموس بيانات IBM DB/DC (أعيدت طباعتها بإذن من DB/DC Data Dictionary Guide من DB/DC Data Dictionary Guide من IBM)

خواص وسمات : Characteristics and Features

لقد ناقشنا حتى الآن المكونات الأساسية لقاموس البيانات: ملفات بيانات تصف بيئة تشغيل البيانات وعلاقات بين كينونات تشغيل البيانات في هذه الملفات. وفي هذا القسم سوف نفحص بعض الاختلافات الموجودة بين قواميس البيانات.

بعض قواميس البيانات عبارة عما يسمى بالوقوف الحر «free standing» ويعني ذلك أنها فئة مستقلة من الملفات الخطية التي يمكن أن تتواجد مع العديد من

نظم إدارة قواعد البيانات والنظم الأخرى. وتصمم قواميس بيانات أخرى لتكون تطبيقات لنظام خاص لإدارة قاعدة بيانات وتتفق ملفاتها مع تكوين ملف نظام إدارة قاعدة البيانات. وحتى في هذه الحالة الأخيرة عادة يكون القاموس قادراً على تخزين بيانات عن كينونات تشغيل البيانات غير الموجودة مع نظام إدارة قاعدة البيانات. وسواء كان القاموس من النوع الأول أو الثاني فتحتاج ملفات قاموس البيانات الى نفس الحماية التي تحتاجها الملفات الأخرى: الأمن والسلامة والنسخة الاحتياطية والاستعادة. فإذا كانت الملفات مبنية على نظام لإدارة قاعدة بيانات فيمكن تطبيق آليات حماية ملف نظام إدارة قاعدة البيانات عليها. فإذا لم يتحقق ذلك (في حالة الوقوف الحر). فيجب أن يقدم نظام قاموس البيانات هذه الآليات بنفسه.

قاموس البيانات، أو بصورة أكثر تحديداً التطبيقات التي تستخدمه، يمكن أن تكون ذات طبيعة تشغيل دفعات أو تشغيل خط مفتوح. ومن الواضح أن بعض التطبيقات التي تتطلب استجابة مباشرة سريعة لا يمكن أن تنفذ في بيئة التشغيل بالدفعة. وهناك تطبيقات أخرى مثل تطبيقات انتاج التقارير يمكنها ذلك وبأخذ ذلك خطوة أخرى للإمام فلا يحتاج قاموس البيانات أن يتواجد كجزء من ملفات الكمبيوتر على الإطلاق. فيمكن أن يحتوي على بطاقات مثقبة أبعادها 3 بوصة في 5 بوصة موضوعة في أحد أدراج التجميع الخاصة بها إلا أنه ينفذ عملياً كأحد تطبيقات الكمبيوتر. ونفترض في هذا الكتاب أن قواميس البيانات تشمل ملفات كمبيوتر وتطبيقات كمبيوتر كذلك كما أنها لها مقدرات اتصال في بيئة الخط المفتاح بصفة عامة.

وكما سبق لنا القول فإن بعض قواميس البيانات لديها إمكانية إضافة أنواع كينونات إضافية أبعد من الأنواع النمطية التي تدعمها. وتختلف مثل هذه الإمكانيات بالنسبة إلى عدد أنواع الكينونات التي يمكن إضافتها وبالنسبة للخواص التي يمكن أن تصاحب الكينونات (في بعض الحالات يجب أن تختار الخواص من قائمة خاصة وفي حالات أخرى يمكن للمستفيد أن يصف أي خواص ويستخدمها) وكذلك في الدرجة التي يمكن بها حفظ العلاقات بين الكينونات الجديدة والقديمة.

صيغ المدخلات: Input Forms

هناك أربع طرق مميزة لتحميل البيانات داخل القاموس بصيغ الدفعات أو من خلال أوامر خط مفتوح فردية أو خط مفتوح بأشكال كاملة للشاشة أو عن طريق الاستخلاص من نظم البرامج والأوامر الموجودة بالفعل. وتصف الأقسام التالية كيفية عمل هذه الطرق.

صيغ الدفعات:

العديد من القواميس لديها إمكانية تسمح لها بتنفيذ مدخلات القاموس لملأ صيغ على ورق تصف المحتويات المعطاة للقاموس. والبيانات الموجودة على هذه الصيغ يتم إدخالها كلها بعد ذلك مرة واحدة على هيئة دفعة واحدة. وعلى سبيل المثال يمكن أن يكون هناك صيغة خاصة للحقول والتي تشمل مكاناً لاسم الحقل ونوعه وطوله وما إلى ذلك. ويمكن أن تملأ هذه الصيغة لأي عدد من الحقول في بيئة تشغيل البيانات التي يراد إدخال وصفها داخل القاموس. وبعد ذلك يمكن إدخالها جميعها في بيئة تشغيل الدفعة مع تجميع ملف الحقول في القاموس. وبينما يبدو عمل أي شيء بنظام تشغيل الدفعة أنه طريقة قديمة إلا أنه يتميز بأنه لا يرهق النظام في نقاط تحميل الذروة (بافتراض أن تشغيل الدفعة يحدث أثناء الساعات التي لا يعمل بها الجهاز. ليلاً مثلاً) وأنه يسمح للعاملين الذين ليس لهم تداخل مع القاموس بأداء العمل.

أوامر الخط المفتوح:

الطريقة الثانية لإدخال البيانات هي بأوامر مكتوبة في أسطر فردية. مثال ذلك السطر التالي:

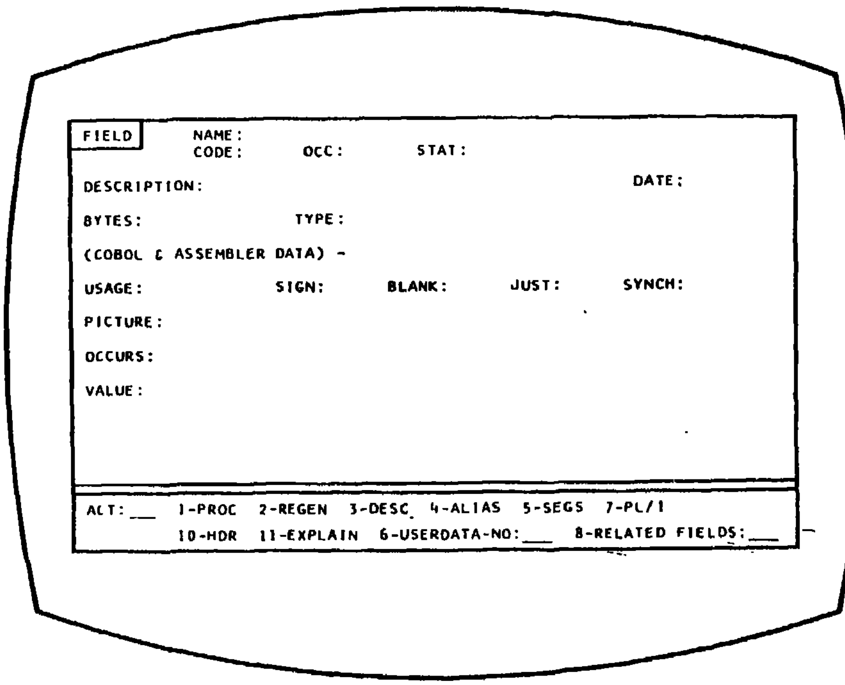
A E EMPNO Type = N Bytes = 6

عندما يتم إدخاله عن طريق نهاية طرفية في بيئة متداخلة فإنه يعني إضافة (A) حقل (أو عنصر (E)) يسمي EMPNO في ملف الحقول ويحدد أن النوع عددي (N) وأن طول الحقل هو 6 بايت. ومن الواضح أن يكون من المرهق جداً إدخال كم كبير من بيانات القاموس باستخدام هذه الطريقة. إلا أنه بالنسبة

لكميات البيانات الصغيرة اللازمة لإجراء تغييرات على البيانات الموجودة فعلاً فـإن هذه الطريقة مفيدة جداً.

أشكال كاملة للشاشة:

طريقة المدخلات الثالثة للقاموس تشمـل عرض شـاشات سـابقة التشكيـل على نهاية طرفية مرئية. ويبين شكل 11.6 مثـل هذه الشـاشة لإدخـال بيانـات عن حقول داخل ملف الحقول. لاحظ أن الشكل يشمل مكاناً لإدخال أسـماء الحقول وأطوالها وأنواعها وما إلى ذلك. ويمكن إدخال وصف القاموس لحقل بسرعـة كبيرة بواسطة شخص مدرب يراقب بدقة الحقول المختلفة الموجودة عـلى الشاشـة سابقـة التشكيل. وبعض قواميس البيانات لديها تجميع أكثر اتسـاعاً لمثـل هذه الشـاشات ويمكن في واقع الأمر ربطها بحيث يمكن لشـاشة معينـة أن تقود إلى شـاشة أخـرى



شكل 11.6

شاشة سبق تشكيلها لمدخلات الحقل من قاموس بيانات IBM DB/DC (أعيدت طباعتهـا بإذن من DB/DC Data Dictionary Guide من IBM)

وتنتج علاقة اتصالات بين محتويات القاموس التي يجري عليها التشغيل. مثال ذلك الشاشة سابقة التشغيل المستخدمة في إدخال معلومات داخل القاموس لسجل والتي تشمل مكاناً للحقول التي توجد في هذا السجل. إدخال أسماء الحقول في شاشة تعريف السجل ينتج عنه علاقة بين السجل وحقوله. ومع الأوامر المناسبة يسمح كذلك للمستفيد بالانتقال من هذه الشاشة إلى شاشة تعريف الحقول لإدخال معلومات تفصيلية أكثر من حقول السجل (كما في شكل 11.6).

الاستخلاص:

الطريقة الرابعة لإدخال البيانات هي أكثر الطرق غرابة فتشمل مجموعات القواميس المختلفة برامجاً تقرأ مجموعات مراقبة قاعدة بيانات ومواصفات قسم البيانات من برنامج الكوبل وما يشبه ذلك، وتستخلص بيانات عنها يجب أن تنتمي إلى القاموس وتدخل البيانات في القاموس. وعندما تستخدم مثل هذه البرامج استخداماً مناسباً فيمكنها أن تمثل وفراً كبيراً واضحاً في الوقت والطاقة بالنسبة للثلاثة طرق الأخرى وكذلك وفراً كبيراً في الأعمال اليدوية وفي صيغة إدخال القاموس. تذكر بالرغم من ذلك أنه بالنسبة لمجهودات إعداد التطبيق الجديد فإن الأشياء هذه مثل مجموعات التحكم ومواصفات بيانات البرنامج يتم إنتاجها بعد تجميع القاموس ببيانات أساسية خاصة بالتطبيق.

صيغ المخرجات: Output Forms

قواميس البيانات قادرة على إنتاج أنواع مخرجات عديدة. بعض هذه المخرجات تكون في صورة تقارير للاستهلاك الآدمي وبعضها الآخر في صورة نظم برامج لاستخدام النظام.

ومن صيغ المخرجات التي تعد للاستخدام الآدمي تمثل القوائم البسيطة المطبوعة أوضح هذه الصيغ. مثل هذه القوائم يجب أن تشمل كل إمكانيات نوع معين من أنواع الكينونات (مثل قائمة بكل الحقول في بيئة تشغيل البيانات مع خواصها). ويمكن أن تشمل فئة جزئية منها فقط مع كل أو جزء من خواص كل منها. وهناك نوع آخر من التقارير المطبوعة وهو القوائم ذات الأدلة المتقاطعة. ويمكن أن تأخذ هذه التقارير صيغة قائمة بكل الحقول الموجودة في سجل معين أو

قائمة بكل المبرمجين المسموح لهم بالاتصال بملف معين وما إلى ذلك. وكمثال لتقرير قاموس (من قاموس بيانات DB/DC من شركة IBM) فإن شكل 11.7 يوضح هرمية DL/I وقطاعاتها والحقول الموجودة في كل قطاع.

```
REPORT DBS (D,P,DDSPDBS,0) NAME=HIER ;

  * * * * * * * *          DB/DC DATA DICTIONARY REPORT        05/06/80  00:24:33
* * * * DBS => SEG => DTE REPORT * * * *                                 PAGE:0001

 DBSNAME: DP DDSPDBS 0
     SEGMENTS: DA DBS 0                          LEVEL:1        PARENT:
                    ELEMENTS: DA DBSNAME 0              START: 1     /0     SEQ: U
                              DA DBSPFX1 0              START: 1     /0     SEQ: G
                              DA DBSPFX2 0              START: 2     /0     SEQ: G
                              DA DBSNMFL 0              START: 3     /0     SEQ: G
                              DA DBSSFX 0               START: 34    /0     SEQ: G

                DA DBSDTE 0                      LEVEL:2        PARENT:DA DBS          0
                    ELEMENTS: DA DTENAME 0              START: 1     /0     SEQ: U
                              DA DTEPFX1 0              START: 1     /0     SEQ: G
                              DA DTEPFX2 0              START: 2     /0     SEQ: G
                              DA DTENMFL 0              START: 3     /0     SEQ: G
                              DA DTESFX 0               START: 34    /0     SEQ: G

                DA DBSSEG 0                      LEVEL:2        PARENT:DA DBS          0
                    ELEMENTS: DA SEGNAME 0              START: 1     /0     SEQ: G
                              DA SEGPFX1 0              START: 1     /0     SEQ: G
                              DA SEGPFX2 0              START: 2     /0     SEQ: G
                              DA SEGNMFL 0              START: 3     /0     SEQ: G
                              DA SEGSFX 0               START: 34    /0     SEQ: G
                              DA PPNAME 0               START: 35    /0     SEQ: G
                              DA PPPFX1 0               START: 35    /0     SEQ: G
                              DA PPPFX2 0               START: 36    /0     SEQ: G
                              DA PPNMFL 0               START: 37    /0     SEQ: G
                              DA PPSFX 0                START: 45    /0     SEQ: G
                              DA SEGLVL 0               START: 46    /0     SEQ: G
                              DA SEGDSGNO 0             START: 47    /0     SEQ: G
                              DA PAIREDLC 0             START: 48    /0     SEQ: G
                              DA PCPNTRS 0              START: 49    /0     SEQ: G
                              DA SEGTYPE 0              START: 50    /0     SEQ: G

                DA SEGPATRB 0                    LEVEL:3        PARENT:DA DBSSEG       0
                    ELEMENTS: DA FREQ 0                 START: 1     /0     SEQ: G
                              DA HPNTRS 0               START: 5     /0     SEQ: G
                              DA TPNTRS 0               START: 6     /0     SEQ: G
                              DA LTPNTRS 0              START: 7     /0     SEQ: G
                              DA LPARNT 0               START: 8     /0     SEQ: G
                              DA IRULE 0                START: 9     /0     SEQ: G
                              DA DRULE 0                START: 10    /0     SEQ: G
                              DA RRULE 0                START: 11    /0     SEQ: G
                              DA WHRRULE 0              START: 12    /0     SEQ: G
                              DA COMPRTN 0              START: 13    /0     SEQ: G
                              DA CCRTNOPT 0             START: 21    /0     SEQ: G
                              DA COMPINIT 0             START: 22    /0     SEQ: G
                              DA CTR 0                  START: 23    /0     SEQ: G

                DA SEGPCHLD 0                    LEVEL:3        PARENT:DA DBSSEG       0
                    ELEMENTS: DA PCSGNAME 0             START: 1     /0     SEQ: G
```

شكل 11.7

تقرير قاموس يصف هرمية DL/I عن طريق قطاعاتها وحقولها IBM DB/DC Data Dictionary (أعيدت طباعتها بإذن من OS/VS DB/DC Data Dictionary Applications Guide من شركة IBM)

وهناك نوع آخر من المخرجات التي يستخدمها الآدميون وهو الاستفسار عبر الخط المفتوح. وأحد أمثلة ذلك هو استخدام الشاشات سابقـة التشكيل التي سبق الكلام عنها في المدخلات وذلـك لفحص محتويـات ملفات القـاموس. وبـدلاً من استدعاء وصف أحد الكينونات بشاشـة سابقـة التشكيل بهـدف الإضافـة إليها أو تغييرها فيمكن أن نريد فحصهـا ببساطـة فقط. ويمكن كذلـك أن يقدم القـاموس إمكانية البحث عن كلمات رئيسية. مثال ذلك إذا كان كـل حقل في ملف الحقـول له وصف نص مخزن معه فيمكن أن نريد البحث عن كلمات رئيسية معينـة في هذه الأوصـاف لإيجاد إذا كـان أحد الحقـول الخاصـة الذي يكـون اسمـه غـير معـروف موجوداً أم لا.

وربما تكون مخرجات القاموس الأكثر شيوعاً والتي تستخدم في تشغيل النظام نفسـه هي إنتاج مجمـوعـات التحكم التي يتـطلبهـا نـظام إدارة قـاعـدة البيـانـات المستخدم. مثل مجموعات التحكم هذه حرجة بالنسبـة لعمل النـظام. وأقل خـطأ يحـدث فيها يتسبب في مشـاكـل تشغيـل رئيسيـة بصفـة عـامـة. لكن إفـرض أن المعلومات عن الملفات والسجلات والحقول وما إلى ذلك الموجودة في القاموس قـد تم إدخالها بعناية مع فحصها بعناية كذلك وإن نظم البرامـج موجـودة وقادرة عـلى أخذ هذه المعلومات في القاموس وإنتاج مجموعات تحكم نظام إدارة قاعدة البيانات منهـا. وعلى هـذا فبينما يـظل هناك فـرصة للخـطأ في التعليمات المعـطاة لمجموعـة التحكم التي تنتج نظام برامج فسوف تقل الفرصة الكلية للخطأ في إنتاج مجموعـة التحكم.

هناك صيغة أخرى لمخرجات القامـوس وهي (مثل حـالة مجمـوعة التحكم) معكـوس أحد طـرق مدخـلات القامـوس. وكما هنـاك برامج يمكنها قـراءة أجـزاء وصف البيانات للبرنامج واستخلاص مدخـلات ووضعها داخـل القامـوس فهناك كذلك برامج يمكنها أخذ المحتـويات من القـاموس وإنتـاج عبارات وصف بيـانات البرنامج مثل ما هو مستخدم في جزء بيانات بـرنامـج الكوبـل. ومن الواضـح أن مثـل هذه الـبرامج، كـما هو الحـال في المدخـلات، تقدم مـع توجيهـات تحدد أي محتويات القاموس يجب أن تستخدم وكيفية وضعها بجانب بعضها البعض.

ويمكن أن تستخدم المواد الموجودة في قاموس البيانات كذلك داخـل وسائـل

أخرى. ومن أمثلة ذلك وسائل تصميم قواعد البيانات التي تتطلب حقولاً وعلاقات بينها كمدخلات وكذلك كاتبات التقارير.

التنفيذ: Implementation

المزايا:

يمكن أن تكون قواميس البيانات ذات استخدام هام للعديد من الأفراد في بيئة تشغيل البيانات. وسوف نفحص في هذا القسم القاموس من وجهات نظر هؤلاء الناس بالنسبة إلى استخداماتهم الخاصة له.

من الواضح أن أكثر المستفيدين من القاموس هو مدير إدارة البيانات والعاملين معه. وكما سيتضح لنا بتفاصيل أكثر في فصل لاحق فإن إدارة البيانات data administration تحمل بإدارة managing البيانات في أقل بيئة. وكما سبق أن رأينا فإن قاموس البيانات هو المكان المناسب لتوثيق وجود الملفات والحقول والتقارير وما إلى ذلك والعلاقات الموجودة بينها وحفظ هذه المعلومات لتقع في نطاق عمل إدارة البيانات. بالإضافة إلى ذلك فعادة ما تحمل مجموعات إدارة البيانات بعمل إنتاج مجموعات تحكم لنظام إدارة قاعدة البيانات. وكما سبق أن حددنا فأفضل طريقة لأداء هذا النشاط يكون باستخدام قاموس البيانات. وعادة يكون العاملون في إدارة البيانات مشمولين على أحد المستويات العديدة بتصميم قاعدة البيانات. ويمكن أن يكون هناك حاجة أو ارتباط لتكوين قاعدة البيانات والحقول وما إلى ذلك المخزنة بالفعل في القاموس من تطبيقات سابقة مع التصميم الجديد. وعلى هذا فيمكن أن يكون القاموس موفراً في الوقت. والأكثر من ذلك مع إعداد التصميم الجديد فإن القاموس يكون المكان المناسب لتخزينه وذلك لكل من أغراض التوثيق ولأغراض انتاج مجموعة التحكم مستقبلياً.

وهناك استخدامات أخرى مختلفة للقاموس بواسطة إدارة البيانات ومعظمها مبني على بيانات من نوع العلاقات وفيما يلي ثلاثة من الأمثلة الممكنة. إذا اشتركت برامج عديدة في ملف واقترح مالك أحد هذه البرامج تغييرات في الملف فيجب على إدارة البيانات أن تمس الأساس مع ملاك البرامج الأخرى لتحديد إذا كان التغيير في الملف سوف يؤثر على برامجهم أم لا (ويسمى ذلك بتأثير

التغيير). يمكن تحقيق ذلك باستخدام بيانات علاقية بين كينونات الملف والبرنامج في قاموس البيانات فالملف المعطى يكون مصاحباً له عدد من البرامج. فإذا تسربت بيانات سرية من الشركة فيمكن أن يبدأ البحث باسترجاع إدارة البيانات معلومات العلاقات بين العاملين والملفات لمعرفة من (على الأقل من خلال القنوات المعتادة) المسموح له بالاتصال بالبيانات. وإذا كان هناك عمل يؤدي في التخطيط طويل الأجل ويراد عمل مراجعة للبيانات المستخدمة حالياً في التطبيق الذي تعاد دراسته فيمكن لإدارة البيانات أن تستخدم العلاقة بين الحقول وتكوين قاعدة البيانات في هذا الغرض.

يمكن لمحللي النظم ومصممي البرامج استخدام قاموس البيانات بطريقتين أساسيتين. أحد الاستخدامات هو كمصدر معلومات عن أي الحقول والملفات وما إلى ذلك الموجودة فعلاً في بيئة تشغيل البيانات والتي يمكن أن تكون مطلوبة في إعداد تطبيق جديد يجري العمل عليه. وإذا كانت البيانات اللازمة لنظام جديد موجودة بالفعل فيكون على ذلك التطبيق الجديد قادراً على استخدامها. فإذا كان هناك تكوينات قواعد بيانات موجودة فعلاً يمكن أن يضاف اليها لإشباع حاجة التطبيق فيمكن أن يكون هذا وفراً كبيراً في هذه الحالات والحالات الشبيهة يكون القاموس عبارة عن مخزن للمعلومات يجري فحصه.

الاستخدام الآخر للقاموس بالنسبة لمحللي النظم ومصممي النظم هو كوسيلة توثيق للمعلومات الجديدة التي تنتج لمجهود تطوير تطبيقاتها. ففي بيئة تطوير معقدة للتطبيقات يبدأ محلل النظم العملية بإدخال بيانات غير طبيعية عن الحقول (على الأقل اسم الحقل) كما تظهر أثناء العمل. ويمكن بعد ذلك لمصمم النظام أن يضيف معلومات طبيعية عن الحقول ومعلومات عن الملفات وتكوينات قاعدة البيانات وما إلى ذلك. وعلى هذا يكون لدى مطوري التطبيقات وسيلة توثيق ويكون القاموس طريقة طبيعية لانتشاره مع البيانات الخاصة بالتطبيقات الجديدة.

وأخيراً هناك ميزة قاموس البيانات للمستوى الأعلى لإدارة تشغيل البيانات ولمستوى الإدارة العليا للمؤسسة ككل. ومع تخلل تشغيل البيانات في الأوجه المتزايدة وفي مستويات الإدارة فتزداد أهمية فهم الإدارة لطبيعة البيانات في نظامها.

ومن المهم من وجهة نظر فهم كيفية عمل المنطقة اعتبار البيانات مورداً يوجه العاملين في المنظمة. وبينما يدير أفراد إدارة البيانات قاموس البيانات مع وجود مثل هذا النوع من البيانات فيه فإن الإدارة هي المستفيدة الإفادة الكبرى من ذلك.

التكاليف:

هناك عدد من التكاليف التي تصاحب وجود قاموس البيانات وكما هو الحال مع أي تطبيق آخر من تطبيقات الكمبيوتر فهناك تكاليف برامج القاموس نفسها وتكاليف تنفيذها وصيانتها وهناك تكاليف تصاحب تعليم الأفراد كيفية إدخال بيانات في القاموس وتعليم أفراد آخرين كيفية الاستفادة من استخدام القاموس.

والتكلفة التي تثير معظم الجدل بالنسبة للقاموس نشمل المجهودات الأساسية والمستمرة في تطبيق مع البيانات ونشره معها. عندما يتم إدخال قاموس بيانات في بيئة تشغيل بيانات فإنه ينتظره سنوات وسنوات من تركيم كينونات تشغيل البيانات كمحتويات له. ومحاولة إدخال كل هذه المادة داخل القاموس مرة واحدة يمكن أن يكون عملاً غير طبيعياً. فعادة ما تبنى ملفات القاموس تدريجياً بقطع مناسبة من البيانات تنتج كنتيجة لمجهودات تطوير التطبيقات الجديدة. ولدينا الكثير لقوله عن هذا الموقف بإيجاز ولكن يكفي القول الآن أنه سواء بذل مجهود أساسي كبير أو أن البناء التدريجي للقاموس هو الذي يحدث أو كلا منهما مع بعضهما فهناك مستوى معنوي من التكلفة المصاحبة لكل عملية من هذه العمليات.

متطلبات النجاح:

المبادلة بين التكلفة والمنفعة لقواميس البيانات هي مبادلة محيرة وهناك عدد من العوامل التي تؤثر على الحصول على تشييد ناجح للقاموس.

من الواضح أنه يجب أن تدعم الإدارة فكرة تشييد قاموس البيانات. فيجب أن توافق على المبالغ المطلوبة لنظم البرامج الأولية وتحديد الأفراد والعاملين لأداء ذلك العمل. كما يجب أن تكون الإدارة متمتعة بالصبر بالسماح بمرور الوقت أثناء انتشار القاموس مع البيانات حتى يمكنه أن يحقق ما هو متوقع منه.

وهناك استخدامات معينة للقاموس لا تتطلب تواجد كم هائل من البيانات

في ملفات القاموس. وأحد أمثلة ذلك هو تطبيق انتاج مجموعة تحكم قاعدة بيانات. فالبيانات الوحيدة التي يجب أن تكون موجودة في القاموس هي البيانات التي يجب أن توجد في مجموعات التحكم هذه. ولكي يمكن جعل القاموس منتجاً بأسرع ما يكون فعادة ما يتم اقتراح أن يكون انتاج مجموعة التحكم هذه هي إحدى أولى تطبيقات القاموس الجديد للسبب سالف الذكر.

والوجه الآخر للعملة هو أنه هناك استخدامات أخرى للقاموس تتطلب مجهوداً أكبر أو مكلفاً في إدخال البيانات داخل القاموس قبل إمكانية التمتع بمميزاته. خذ على سبيل المثال حالة محلل النظم الذي يبحث داخل القاموس لمعرفة إذا كانت بعض البيانات اللازمة في تطبيق جديد يعد حالياً موجودة بالفعل في بيئة تشغيل البيانات أم لا. وفيما عدا أن كل - أو على الأقل معظم - التطبيقات الموجودة تكون معكوسة في القاموس فلن يحصل محلل النظم على ثقة كبيرة في البحث في القاموس تأتي بالرد بأن البيانات غير موجودة not found فربما تتواجد البيانات إلا أنها لم تودع في القاموس بعد.

وعلى هذا فالطريق الى تشييد قاموس بيانات ناجح هو طريق تدريجي. أولاً تأكيد الإدارة على الفكرة الخاصة بالحصول على قاموس بيانات يجب أن يحدث. وبعد أن يشيد القاموس يجب أن يكون إدخال البيانات مخططاً تخطيطاً جيداً ويحدث بصورة تقدمية. وعادة ما يكون الطريق المختار هو انتشار القاموس مع البيانات المستخلصة من تطبيقات تشغيل البيانات الجديدة التي تعد. والاستخدام الفوري لمثل هذه البيانات مثل انتاج مجموعات التحكم هو أحد أسباب عمل ذلك. إلا أنه هناك سبب آخر وهو كفاية القاموس لتوثيق التطبيقات الجديدة وهو شيء يمكن عمله بطريقة أو بأخرى في أي حالة. وهناك سبب ثالث وهو نقطة عاطفية وهي أنه يجب على التطبيق الجديد أن يصاحبه أحدث التقنيات. وأخيراً حيث أن هذه هي بداية ملفه فيمكن أن يظل هذا التطبيق تحت التطوير فترة طويلة وهذا يؤدي إلى استخدام أطول للقاموس.

بالإضافة إلى ذلك يمكن إعداد جدولة لإحضار تطبيقات المنظمة الموجودة فعلاً داخل عالم القاموس وذلك في فترات زمنية معقولة. مرة أخرى لأداء هذا

العمل يجب أن تثق الإدارة في المنافع طويلة المدى للقاموس وتدعم الموارد اللازمـة وتتحلى بالصبر بالنسبة للعائد منه.

كما أن المتطلبات الأخرى لتشييد قاموس ناجح تدور كـذلك حـول الأفراد. فهؤلاء الأفـراد سـواء كـانـوا عـاملين في أدرة البيـانـات أو محللي النـظم أو مـبرمجين يجب أن يكـونوا متعلمـين وملمين بكيفيـة إدخال بيـانات داخـل القامـوس وكيفية استخـدامه. ويجب أن يكـونوا ملتصقـين بالنمـطيـات وتـرتيبـات الأمن في تنفيـذ اجـراءات تشمل القـاموس. وأخيـراً فيفضـل أن تثق الإدارة في قيمـة القـامـوس وكـذلك يكـون لديهـا اتجاه قـوي لاستخدامـه وتعديـل مســاره إذا حـدث تغيـير في الأوقات المحددة له.

قواميس نشطة وأخرى سلبية : Active VS. Passive Dictionaries

هناك تمييز يتزايد اهتمامه في قواميس البيانات وهو قـواميس البيانـات النشطة أو المتكاملة وقواميس البيانات غير النشطة أو السلبية. في الواقـع لا يوجـد تعريف واحد مقبول لهذه الاصطلاحات وهناك مدى وظائف واسع متصل بها.

وفي الأســاس فإن القـاموس السلبي هـو القـامـوس المستخـدم في أغـراض التوثيق فقط. ويتم إدخال البيانات الخاصة بالحقول والملفات والأفراد وما إلى ذلك داخل القاموس مع الإشارة إليها تقاطعياً. والمعلومات في صورة القـوائم المطبـوعة والاستفسارات من النهايات الطرفية عن محتويات القامـوس يتم تشغيلها كلما كـان هناك حاجة لذلك. والقاموس السلبي هو تطبيق احتواء ذاتي ببساطة ومجموعة من الملفات المستخدمة في توثيق بيئة تشغيل البيانات.

لكن مـاذا عن المدى الـواسع لـلأشياء الأخـرى المشمولـة؟ حيث أن معظم القواميس تكون مصـاحبة بقـرب شديـد في التكوين والاستخـدام أو في كل منهـما لنظام إدارة قاعدة البيانات فلنعتبر أحد القواميس في هـذه البيئة. إحـدى الأنشطة التي نـظرنا لهـا بالفعـل التي تحضر القامـوس ليكون قـريباً إلى عمـل بيئـة قـاعـدة البيانات هي انتاج مجموعات تحكم نظام إدارة قاعدة البيانات من البيانات الموجودة في القاموس. ويمكن إضافة البرامج التي تنتج تعريفات بيانات لبرامـج تطبيقـات من البيـانات المـوجودة في القـاموس والعكس مـع الـبرامـج التي تنشر القـامـوس مـع

البيانات المستخلصة من شفرة المصدر لبرنامج التطبيق. كل أنشـطة القامـوس التي تعد في بيئة الخط المغلق هذه والتي تضيف المزيـد إليه أو تستخلص منتجـات نظام منه تشكل خطوة أبعد من القاموس السلبي البسيط.

هنـاك إمكانيـة أخرى تجعـل من القامـوس نشـطاً بعض الشيء وهي تنفيـذ متطلبات requirement الوظائف مثـل التي سبق ذكرهـا. مثال ذلـك يمكن لنظام إدارة قاعدة البيانات أن يجبر كل استدعاءات بـرامج التـطبيقات لقـاعدة البيـانات على التقاط التفـاصيل الـطبيعية لبيـانات التـطبيق من القامـوس عند وقت الـترجمة compile time. وعلى هذا عنـد إعطاء اسم حقل بواسطة أحد برامج التطبيقات فيمكن للقاموس عند وقت الترجمة تقديم معلومات عن مكان تخزين الحقل وكيفية تخزينه والتي يمكن أن تترجم داخل شفرة التنفيذ.

هنـاك خطوة أخـرى لأعلى في هـذا الخط من التفكير وهي حصـول برنـامج قاعدة البيانات على معلومات من أو مـراقبته جـزئياً بـواسطة (طبقـاً لكيفية نـظرك إليه) قاموس البيانات أثناء وقت التنفيـذ. ويمكن أن يمرر القـاموس معلومـات عن تكوين البيانات عند وقت التنفيذ لم يحدث لها ترجمة داخل برنامج التنفيذ للتـطبيق أو كمثال آخر يمـكن أن يقـدم معلومات أمن عن أي المستفيـدين أو أي البرامـج يسمح لها بالاتصال بملفات محددة.

وبـدون تدخـل آدمي فكل القـواميس النشطة يمكن أن تكـون قـادرة عـلى حصر بـرامج تـطبيقات قـاعدة البيـانات التي تشيـد وتنفذ لاستخـلاص معلومات بصورة تلقائية عنها ليست موجودة بالفعل في القاموس.

وقد اعتبرت قواميس البيانات الأصلية أنها وظيفية لأحد التطبيقات الإضافية فقط في بيئة قاعدة البيانات. ومع تقـدمنا في عـالم القواميس النشـطة المتزايـد نجد العـديد ممن يشعـرون أن القواميس سـوف توجـد في قلب بيئة نـظم إدارة قواعـد البيانات في المستقبـل. وبمعنى معين سيكـون القامـوس هو المـراقب المركـزي لبيئة نظام إدارة قاعـدة البيانـات يديـر ملفات التـطبيقات وعمليـات الأمن والاحتياطي والاستعادة وكذلك نمطيات البرمجة.

References مراجع:

Allen, F. W., Loomis, M.E.S., and Mannino, M. V., «The Integrated Dictionary/ Directory System», Computing Surveys, vol. 14, no. 2, June, 1982, pp. 245 - 286.

Leong-Hong, B.W., and Plagman, B.K., Data Dictionary/ Directory Systems, Wiley, New York, 1982.

Ross, R.G., Data Dictionaries and Data Administration, AMACOM, New York, 1981.

Van Duyn, J., Developing a Data Dictionary Systems, Prentice-Hall, Englewood Cliffs, NJ, 1982.

Questions and Exercises أسئلة وتمارين:

١ ـ صف بكلمات من عندك قاموس البيانات ولماذا يكون هناك حاجة إليه.

٢ ـ بنفس نمط ملفات القاموس الموجودة في شكل 11.3 اعمل ملف قاموس لتقارير. ماذا يمكن أن يكون معنى العلاقات بين ملف التقارير هذا والملفات الموجودة في شكل 11.3؟

٣ ـ بنفس النمط الموجود في ملفات القاموس الموجودة في شكل 11.3 اعمل ملف قاموس للعاملين. ماذا يمكن أن يكون معنى العلاقات بين ملف العاملين هذا والملفات الموجودة في شكل 11.3؟.

٤ ـ بنفس النمط الموجود في ملفات القاموس الموجود في شكل 11.3. اعمل ملف قاموس للبرامج. ماذا يمكن أن يكون معنى العلاقات بين ملف البرامج هذا والملفات الموجودة في شكل 11.3؟.

٥ ـ صف كينونات ممكنة لقاموس بيانات لم تذكر في هذا الفصل. ما العلاقات التي يمكن أن توجد بينها وبين الكينونات النمطية؟

٦ ـ أسرد مجموعة شاملة من الخواص لكل من:

أ ـ ملف الحقول.

ب ـ ملف الملفات.
جـ ـ ملف العاملين.
د ـ ملف البرامج.

٧ ـ اختر خمس من العلاقات المبينة في شكل 11.3 (عرفها بواسطة ملفات القاموس التي تصلها) وحدد معنى كل منها.

٨ ـ ما مميزات وعيوب قواميس بيانات الوقوف الحر وقواميس البيانات المحددة لنظم إدارة قاعدة بيانات؟

٩ ـ صف ميزة معنوية وعيب معنوي لكل من الأربع طرق الخاصة بمدخلات قاموس البيانات التي سبق ذكرها في هذا الفصل.

١٠ ـ صمم شاشة كاملة سابقة التشكيل لمدخلات من نهاية طرفية داخل قاموس بيانات لما يلي:
أ ـ تقرير.
ب ـ ملف.
جـ ـ برنامج.
د ـ أحد العاملين.

١١ ـ صمم شكلاً لتقرير قاموس بيانات لأحد التقارير الذي يصف المعلومات الموجودة في ما يلي:
أ ـ ملف حقول.
ب ـ ملف ملفات.
جـ ـ ملف عاملين.
د ـ ملف برامج.

١٢ ـ صمم تقرير قاموس بيانات لمعلومات أدلة متقاطعة بين:
أ ـ ملف الحقول وملف الملفات.
ب ـ ملف التقارير وملف العاملين.
جـ ـ ملف البرامج وملف الحقول.

١٣ ـ إقنع مدير تشغيل البيانات أنه يجب أن يشيد قاموس بيانات.

١٤ ـصف تكاليف تشييد وتشغيل قاموس البيانات.

١٥ ـصف بكلمات من عندك كيف يمكنك إدخال قاموس بيانات تدريجياً داخل بيئة تشغيل بيانات.

١٦ ـكيف تتغير بيئة تشغيل البيانات مع استخدام قاموس بيانات مرتفع النشاط؟

١٧ ـهل يغير قاموس بيانات مرتفع النشاط من طريقة المبرمجين ومحللي النظم ومشغلي الأجهزة في أداء أعمالهم؟ وضح إجابتك.

١٨ ـدافع عن أو هاجم العبارة التالية «قاموس البيانات النشط هو البؤرة المركزية لبيئة قاعدة البيانات».

الفصل الثاني عشر
مواضيع خاصة ببيئة قاعدة البيانات

Database Environment Issues

مقدمة : Introduction

كما سبق أن اقترحنا في الفصل الرابع يوجد هناك اهتمامات خاصة بالبيانات، يجب معرفتها بغض النظر عما إذا كانت ستخزن هذه البيانات تحت تحكم نظام إدارة قاعدة ببيانات أم لا. هذه الاهتمامات تشمل حفظ سرية البيانات وحماية سلامتها في مواقف التحديد المتزامنة والمقدرة على استعادة البيانات إلى حالة تصحيحة بعد حدوث خطأ في البيانات وعند حدوث تدمير للمحتويات الداخلية لقاعدة البيانات والمقدرة على عمل مراجعة محاسبية للبيانات لمعرفة من الذي اتصل بها أو قام بتغييرها.

مرة أخرى يجب على بيئة قاعدة البيانات أن توفر مواقف لكل هذه المشاكل دون أن يطلبها المبرمجون في كل مرة يظهر تطبيق جديد. ويمكن أن تكون حلول كل من هذه المواقف حلولاً عامة لأي بيانات تتسق مع هيكل بيئة قاعدة البيانات المعطاة. وفي هذا الفصل سوف نلقي نظرة سريعة على أمر من هذه الأمور الأربعة لبيئة قاعدة البيانات.

الأمن : Security

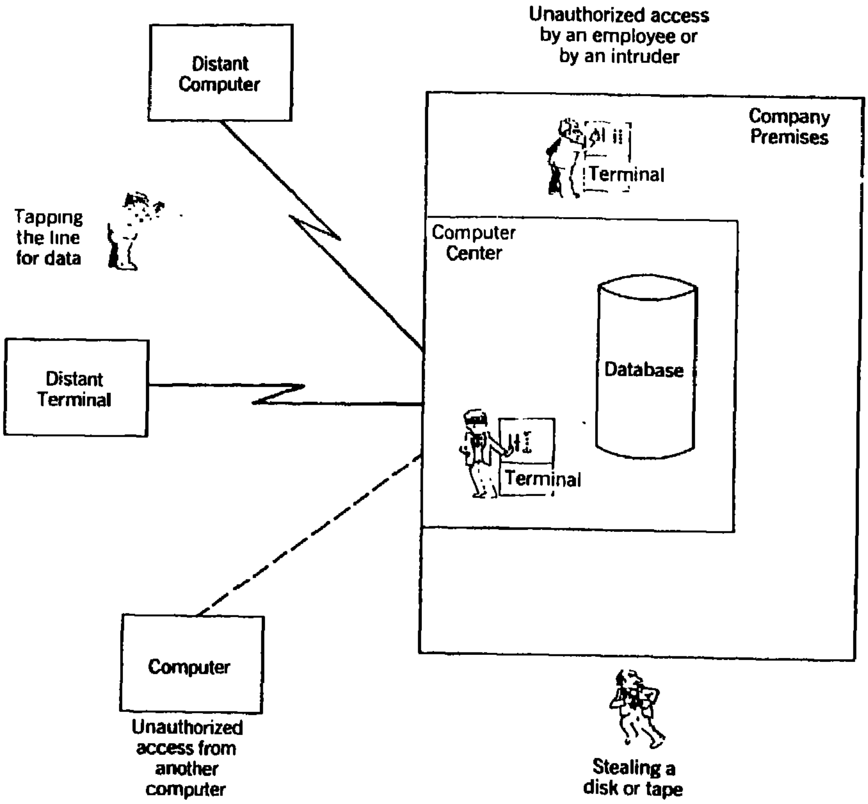
هل من المهم حماية البيانات المخزنة من اتصال غير المسموح لهم بها وإجراء تغييرات غير مصرح بها عليها؟ بكل تأكيد نعم وذلك للعديد من الأسباب. فمع تزايد استخدام الكمبيوتر في حفظ سجلات مجتمعنا أكثر وأكثر كلما ازدادت أنواع البيانات المختلفة وازداد حجمها بسرعة كبيرة والتي يراد تخزينها في ملفات الكمبيوتر بما في ذلك بالطبع قواعد البيانات. أما المنظمة التي بها البيانات أو

الأفراد أو المنظمات التي تصفها البيانات يكون لهم اهتمامات كبيرة بأمن البيانات. ويجب أن تحمي الحكومات بيانات الدفاع الحساسة من التدخل غير المسموح به. كما يجب أن تمنع الشركات منافسيها من رؤية بياناتها المحاسبية. ويجب أن تتأكد البنوك أن الأموال التي في حوزتها والموجودة على هيئة بيانات في الكمبيوتر الآن لا يمكن سرقتها (مثل قيمة حقل تمثل مبلغاً من المال في أحد الحسابات لا يمكن أن يتم الوصول إليه). ويريد الأشخاص أن تظل المعلومات الشخصية التي تحتفظ بها شركات التأمين عنهم سرية. وتستمر القائمة طولاً.

دعنا نبدأ باعتبار أنواع هتك الأمن الممكنة في تسهيلات تشغيل البيانات الحديثة. ربما يكون النوع الأساسي جداً للهتك هو الاتصال غير المصرح به بالبيانات، أي أن شخص معين يرى بيانات غير مسموح له برؤيتها. هناك شيء آخر وهو التعديل غير المصرح به في البيانات، أي أن أحد الأشخاص يغير من قيمة بيانات مخزنة ولا يكون مسموحاً له بعمل ذلك. ومشكلة ثالثة وهي إحدى صيغ التوسع في المشكلتين السابقتين هي التعديل غير المصرح به في البرامج والذي يحدث فيه أن أحد الأشخاص يغير من البرنامج بحيث أن البرنامج يسترجع أو يعدل بيانات بصورة غير قانونية.

وعند ذلك يصبح السؤال سؤالاً عن التفكير في طرق يمكن حدوث مثل هذه التسويات المقصودة للبيانات، شكل 12.1. إحدى طرق الدخول إلى قاعدة بيانات الشركة هي خطوط اتصالاتها والتي تنقل بيانات بين أجهزة الكمبيوتر الخاصة بها أو بين نهاية طرفية وإحدى أجهزة الكمبيوتر الخاصة بها. وهناك طريقة أخرى وهي أنه إذا كان يمكن الاتصال بالكمبيوتر عبر خطوط هاتف عادية وهي ظهور أحد المحتالين على أنه مستفيد مصرح له باستخدام الكمبيوتر. وهذا يعني كما سوف نناقش ذلك بصورة أكثر اكتمالاً فيما بعد الحصول على رقم هاتف الكمبيوتر وانتاج دليل تعريف يميزه الكمبيوتر وإدخال كلمة المرور الصحيحة. وبطريقة شبيهة يمكن أن يكون المحتال قادراً على إدخال المستندات المختفية للشركة واستخدام نهاية طرفية موجودة في الشركة. هناك طريقة أخرى لإمكانية الاتصال بالبيانات وهي سرقة وسط واقعي توجد عليه البيانات بما في ذلك مجموعات الأقراص والمطبوعات. كما يجب أن يؤخذ في الاعتبار كذلك رشوة

العاملين المسؤولين في الشركة. والقول المأثور بأن «السلسلة قوية فقط بقدر قوة أضعف نقطة فيها» يسري هنا.



شكل 12.1

هتك أمن البيانات

لجعل الأمور أكثر تعقيداً فهناك كذلك حالة العامل الأصلي بالشركة المسموح له بالاتصال ببعض بيانات الشركة والذي يحصل على اتصال ببيانات أخرى غير مسموح له بالاتصال بها. ويأخذ ذلك خطوة أخرى للأمام أكثر من هذا يمكن أن يكون هناك موقفاً يحاول فيه أحد العاملين تعديل بيانات مسموح له بالاتصال بها ولكنه غير مسموح له بتعديلها.

ماذا يحضرنا إلى أصل هذا القسم. ما هي أنواع معايير الأمن التي يمكن استخدامها لحماية بيانات مخزنة وبصفة خاصة (ولكن ليست فقط) البيانات

الموجودة في قاعدة البيانات؟ أولاً وقبل أن تعتبر الشركة الوحدات الموجهة لنظم المكونات أو نظم البرامج التي عادة ما تصاحب فكرة أمن البيانات فهناك موضوع الأمن الطبيعي لمستندات الشركة. ويجب ألا يغفل هذا العنصر. فأي شخص في غرفة الكمبيوتر أو في منطقة توجد بها نهايات طرفية أو أي وحدات مدخلات ومخرجات أخرى يجب أن يميز بأنه أحد العاملين بالشركة أو أحد الزوار المعروفين للشركة. ويطبق ذلك على المناطق التي تحتوي على بعض بيانات للشركة كذلك من مكتبات شرائط إلى غرف مطبوعات كما يأمل الإنسان كذلك أن يكون العاملون في الشركة أمناء إلا أن هذا بالطبع من الصعب مراقبته.

بعد ذلك هناك موضوع مراقبة الاتصال بنظام الكمبيوتر. ونفترض هنا أن المحتال أما أنه يكتسب اتصالاً بنهاية طرفية على مستندات الشركة أو أنه لديه جهاز كمبيوتر (ربما ميكروكمبيوتر) ومودم وأنه يعرف رقم هاتف كمبيوتر الشركة. وتتطلب معظم النظم حالياً من الشخص الذي يحاول استخدام الكمبيوتر أن يعرف نفسه بدليل تعريف من عدة رموز ثم يدخل كلمة مرور تسمح له باستخدامه. وعادة تكون أدلة التعريف معروفة بصفة عامة (على الأقل داخل الشركة). أما كلمات المرور فيجب أن تكون سرية بقدر الإمكان ويجب أن تتغير دورياً ويجب ألا تكتب في أي مكان وذلك لتقليل مخاطرة معرفة أي شخص آخر بها. ويجب ألا تظهر كلمات المرور على الشاشة عند كتابتها كما يجب أن ينتجها المستفيد بحيث تقل فرصة نسيان المستفيد لها.

وتوجد كذلك اقتراحات بصيغ أخرى لتعريف المستفيد. إحدى هذه الصيغ هي تخزين معلومات شخصية من المستفيد (مثل اسم زوجته أو ابنته) ويطلبها منه النظام عندما يحاول استخدامه. ولسوء الحظ حيث أن البيانات لا تجدد على فترات قصيرة فتوجد فترة كبيرة تمكن من التعرف عليها مع الوقت. وهناك إمكانيات أخرى للمستقبل وهي اختيارات من خواص طبيعية مثل بصمة إصبع المستفيد أو التوقيع أو الصوت كوسائل لتمييزه والتي أصبحت معقدة بدرجة كافية.

دعنا نفترض الآن أنه تم تمييز شخص معين بواسطة النظام على أنه مستفيد مسموح له باستخدام النظام. قد يكون هذا الشخص مسموحاً له في الواقع باستخدام النظام أو غير مسموح له ولكنه أقنع النظام أنه مسموح له (ربما عن

طريق سرقته دليل تعريف شخص آخر وكلمة مروره). هناك سمات أمن يمكن استخدامها لحماية البيانات بالرغم من مرور شخص معين عبر العقبة الأولى على أنه مستفيد من النظام. باصطلاحات بيئة قاعدة البيانات ربما يكون الشيء الأساسي جداً هو أن كل إتصال بالبيانات يحدث له مراقبة بواسطة نظام إدارة قاعدة البيانات ويجب أن يمر خلال هذا النظام. ويقود هذا إلى عدد من السمات المفيدة. إحدى هذه السمات تشمل بيانات يحفظها النظام تحدد أي المستفيدين (طبقاً لدليل التعريف) مسموح له باسترجاع وبتجديد أي ملفات البيانات وأنواع قطاعات IMS وأنواع سجلات CODASYL وما إلى ذلك. وعلى هذا فإذا كان هناك شخص معين مسموح له باستخدام النظام لكنه حاول الاتصال ببيانات غير مسموح له الإطلاع عليها فلا يسمح النظام بالتشغيل له. وبالنسبة لأنواع بيانات معينة يمكن أن يسمح للشخص أن يطلع عليها ولا يسمح له بتجديدها وعلى هذا فيجب أن يحدث مثل هذا التمييز. وهناك صيغ أخرى عديدة على هذا الموضوع والتي كما هو الحال مع سمات الأمن سالفة الذكر، يمكن أو لا يمكن بناءها في نظام إدارة قاعدة بيانات معين. مثال ذلك يمكن أن تسمح بعض النظم لمستفيدين معينين برؤية بعض وليس كل الحقول الموجودة في حدوثات أنواع سجلات معينة. وبعض النظم تمنع المستفيدين من رؤية بيانات وليس ذلك مبنياً على أنواع السجلات المشمولة فقط بل كذلك على القيم الموجودة في حدوثات محددة لهذه السجلات. وهناك فكرة أخرى وهي السماح للمستفيد برؤية حقلين مختلفين في حدوثات نوع سجل معين ولكن لا يحدث ذلك في نفس الوقت. مثال ذلك قد يكون مسموحاً للمستفيد استخدام ملف أفراد للحصول على قائمة بأسماء العاملين وللحصول على قائمة بكل الرواتب لأغراض العمل الأحصائي، إلا أنه قد لا يكون مسموحا له بمعرفة رواتب أشخاص معينين أي أنه يضبط رواتب شخص معين مع اسمه.

هناك سمة أخرى لحماية البيانات مرة أخرى خاصة بالموقف الذي يميز فيه الشخص بأنه مسموح له باستخدام النظام وهو الاستخدام الإضافي لكلمات المرور. لقد تكلمنا عن استخدام كلمات المرور في حماية النظام من الدخول غير المسموح فيه. إلا أن كلمات المرور يمكن أن تحدد كذلك لبرامج معينة وللملفات

بيانات معينة. ويجب أن يطلب النظام من المستفيد أن ينتج كلمة المرور اللازمة لتنفيذ برنامج معين أو للاتصال بملف معين.

ولسوء الحظ لا يستخدم أي من الأساليب السابقة للحماية ضد السرقة التي لا يحاول فيها المحتال استخلاص البيانات من خلال الكمبيوتر ونظام إدارة قاعدة البيانات. ومثال لهذا الموقف يشمل الحصول على البيانات من خطوط الاتصالات التي تنقلها أو سرقة مجموعة أقراص. وحل هذه المشكلة هو خلط البيانات data encryption. ويشمل اصطلاح خلط البيانات أن البيانات يتم تغييرها حرفاً حرفاً أو بت بت في صورة تشبه Garbled تماماً. ويجب أن تعاد البيانات إلى حالتها الطبيعية ليمكن استخدامها. وعملية الخلط وإعادة الخلط يمكن أن تحدث عندما ترسل البيانات عبر خطوط اتصالات فقط. أو أن تخزن البيانات في صورة مخلوطة على القرص.

ويمكن أن تتراوح عملية خلط البيانات من عمليات بسيطة إلى عمليات معقدة. ومن الواضح أنه كلما كانت العملية بسيطة كلما كان من السهل للمحتال أن يحلها ويعيد فكها لمعرفة البيانات الأصلية. وتشمل عمليات الخلط بصفة عامة استخدام طريقة معينة ومفتاح سري. وأحد الأمثلة البسيطة جداً لخلط الحروف الهجائية هي كما يلي. يتم ترقيم الحروف من A إلى Z ومعها الفراغ بالأرقام من 1 إلى 27. ويضاف لكل حرف يراد خلطه مفتاح سري (رقم معين في هذه الحالة). ويتغير هذا الحرف إلى الحرف الذي يناظر هذا الرقم الجديد. فإذا كان المفتاح السري 4 فإن A (قيمتها 1) تصبح E (حيث أن 5 = 4 + 1 وهو المناظر للحرف E) ويصبح الحرف B حرف F وهكذا خلال الحروف الهجائية حتى الفراغ والذي يصبح حرف D (مع الدوران عند الانتهاء من 27 إلى 1). ويجب أن يكون البرنامج المستقبل ملماً بكل من الطريقة المستخدمة والمفتاح السري بحيث يمكنه أن يعمل مستخدماً هذه الطريقة عكسياً لإعادة خلط أو لفك خلط البيانات.

وبصفة عامة كلما ازداد تعقيد طريقة الخلط كلما كان هناك حاجة لوقت أكبر لعمل خلط وإعادة خلط للبيانات وسوف يبطيء ذلك من العملية في بيئة قاعدة البيانات. والأكثر من ذلك إذا كانت البيانات المخزنة على قرص قد حدث لها خلط قبل تخزينها فكل اتصال بالقرص، سواء كان هناك نقل عبر خطوط

اتصالات أم لا، يتطلب وقتاً لإعادة الخليط. وبعض طرق الخلط، (بما في ذلك طرق للنقل و لمفاتيح كبيرة جداً لأحد الطرق المعتادة بها مفتاح من 56 بت) تم بناؤها في نظم المكونات لأسباب تتعلق بالأداء.

التزامن: Concurrency

يفترض العديد من نظم تطبيقات تشغيل البيانات حالياً وبصفة خاصة النظم التي تعمل داخل بيئة تشغيل قاعدة بيانات أن العديد من الناس يستخدمون هذه النظم وأنهم في حاجة إلى الاتصال بنفس ملفات البيانات في نفس الوقت. ونظم المكونات ونظم البرامج الحديثة قادرة بكل تأكيد على توفير هذه المشاركة في الاتصال بالبيانات. وأحد الأمثلة الشائعة جداً لهذا يوجد في حجز مقاعد الطيران حيث يمكن أن يوجد لدى موظفي الحجز طلبات حجز من عملاء مختلفين لمقاعد على نفس الرحلة ويحدث ذلك في نفس الوقت. مثال آخر لذلك هو تطبيق صناعي أو مخزن حيث يجلس العديد من الموظفين يبحثون في نفس الوقت على نفس العنصر.

عندما لا يحتوي الاتصال المتزامن إلا على استرجاع بسيط للبيانات فلا يكون هناك مشكلة. أما إذا تطلب الاتصال المتزامن اجراء تعديلات على البيانات فإن المستفيدين (أو الأكثر) يحاولا تجديد البيانات في نفس الوقت (أو في نفس الوقت طبقاً لما يسمح به نظام البرامج المتعددة) يكون عندهم فرصة كبيرة أكثر للتداخل مع بعضهما الشيء الذي لا يحدث إذا ما كانا يقومان بعمل استرجاع للبيانات. وهذه هي الحالة بكل تأكيد بالنسبة لحجز مقاعد الطيران والمخازن حيث أن بيع مقاعد الطيران على رحلات معينة واستخدام عناصر موجودة في المخزون يتطلب أن يتم مراجعة لعدد المقاعد أو لعدد وحدات العناصر لتقليلها بالمقدار الذي تم حجزه.

باستخدام حجز مقاعد الطيران كمثال ففيما يلي يوجد مثال لما يمكن أن يحدث. إفرض أنه هناك 25 مقعداً على رحلة طيران Acme Airlines رقم 345 يوم 12 مارس. وتلقى اثنان من موظفي الشركة Mr. Brown و Mr. Green مكالمتين هاتفيتين من عميلين مختلفين في نفس الوقت تقريباً وكل منهما يريد الحجز على هذه الرحلة. ويريد عميل السيد Brown حجز أربعة مقاعد بينما يريد عميل

السيد Green حجز ستة مقاعد. حصل السيد Brown على سجل هذه الرحلة من قاعدة البيانات ولاحظ أنه هناك 25 مقعداً خالياً وبدأ يناقش السعر والتفاصيل الأخرى مع عمليه. وفي نفس الوقت تقريباً استرجع السيد Green نفس السجل (أي نسخة أخرى منه) لعميله. وقام كل من الموظفين بعمل عملية البيع مع تحديدهما لقاعدة البيانات من خلال النهايات الطرفية. وأظهرت العملية الجارية للسيد Brown أنه يجب خصم أربعة مقاعد من المقاعد المتاحة على هذه الرحلة ثم حدد السيد Green أنه يجب حذف ستة مقاعد أخرى.

النتيجة في قاعدة البيانات موضحة في شكل 12.2. ويبين سجل الرحلة رقم 345 يوم 12 مارس الآن أنه هناك 19 مقعداً متاحين. ولكن ألا يجب أن يكون هناك 15 مقعداً متاحاً فقط حيث أنه تم بيع 10 تذاكر؟ نعم إلا أن النقطة هي أنه لم يعرف أي من البائعين أنه هناك بائع آخر يقوم بالبيع على نفس الرحلة في نفس الوقت الذي يفعل فيه ذلك. وعندما طرح السيد Brown أربعة مقاعد أظهر السجل أنه هناك في هذه اللحظة 21 مقعداً متاحاً. إلا أنه عندما طرح السيد Green ستة مقاعد فإنه قام بطرحها من إجمالي 25 وليس 21.

وبهذه المناسبة قد تسأل عن إمكانية قيام اثنين من الموظفين بالبحث عن سجل في نفس الوقت وذلك في ملف حجز مقاعد طيران كبير. هل حاولت أن تحجز على رحلة من القصيم إلى جده بعد انتهاء أحد الفصول الدراسية مباشرة؟ نفس الظروف تحدث في معظم التطبيقات اليومية. وقد تدهش كذلك من إمكانية حدوث نفس نوع الخطأ هذا في تطبيقات تحدث فيها عمليات الاسترجاع والتجديد في أجزاء من الثانية وليس في الدقائق التي استغرقت في المناقشات الهاتفية بين السيد Brown والسيد Green وعميليهما. وكما يجب أن يكون واضحاً فالإجابة محددة جداً: نعم مع ما يتبع البرمجة المتعددة في نظم الكمبيوتر التجارية حالياً.

والحل المعتاد لهذه المشكلة هو أسلوب يعرف بأنه الإغلاق lockout.

يصمم نظام إدارة قاعدة البيانات بحيث أنه إذا بدأت إحدى عمليات التجديد تجري على إحدى قطع البيانات فتغلق أي محاولة لبدء عملية تجديد أخرى على نفس البيانات. وبطريقة أو بأخرى على حسب نوع التنفيذ يتم إعادة جدولة

المحاولة الثانية للتجديد. وعندما يرفع الاستفسار الأول الإغلاق على البيانات فيمكن للاستفسار الثاني أن يغلقها لعملية التجديد الخاصة به. ويمكن أن يتغير مستوى عملية الإغلاق. فالإغلاق على مستوى مرتفع مثل مستوى محتويات ملف يمنع لسوء الحظ الكثير جداً عن الإغلاق على قطعة محددة من البيانات من التعديل أثناء عملية التجديد إلا أنه حل كامل منخفض. أما الإغلاق على مستوى منخفض مثل مستوى الحقل على سبيل المثال لا يمنع بقية قاعدة البيانات إلا أنه حل كامل مرتفع نسبياً.

ولسوء الحظ وكما يحدث عادة فإن إدخال هذه الميزة في حد ذاتها يتسبب في مشاكل أخرى لم يسبق لها الوجود. اتبع الحوار التالي:

دعنا نقول أننا لدينا موقف مخزون تصنيع يجب أن يجد الموظف فيه إذا كان هناك كمية كافية من كل من جزئين متاحين لتلبية أحد الطلبات أم لا. وقد كتب أحد البرامج والذي صمم لقبول معلومات من نهاية طرفية عن الجزئين الذين يبحث عنهما الموظف بما في ذلك الكمية اللازمة. وبعد ذلك يبحث البرنامج في قاعدة البيانات عن سجلات هذين الجزئين ويسترجع الكمية الموجودة حالياً من كل منهما ثم تحدث موافقة من الموظف الموجود عند النهاية الطرفية ويطلب الجزئين لشحنهما مع تقليل الكمية المسجلة بقاعدة البيانات بالقيمة التي سوف تشحن. إفرض أنه هناك اثنان من الموظفين السيد White والسيد Black وكل منهما يطلب كمية من الصواويل والمسامير في نفس الوقت تقريباً. وقد قام السيد White بسرد الصواويل قبل المسامير في استفساره بينما قام السيد Black بسرد المسامير قبل الصواويل في استفساره. وبدء كل منهما استفساره في نفس الوقت تقريباً. وقد حصل استفسار White على سجل الصواويل وأغلقه وفي نفس الوقت حصل Black على سجل المسامير وأغلقه. بعد ذلك حاول استفسار White الحصول على سجل المسامير لكنه وجده مغلقاً بواسطة Black. وفي نفس الوقت حاول استفسار Black الحصول على سجل الصواميل لكنه وجده مغلقاً بواسطة White. وفي هذه الحالة ينتظر كل من الاستفسارين إلى مالا نهاية أن يترك أحدهما السجل الذي يريده الآخر. ويسمى ذلك بإغلاق الموت dead lock أو deadly embrace وهو مبين في شكل 12.3 وهو يحمل في واقع الأمر علاقة قريبة من إغلاق الشبكة gridlock المعروف كمشكلة مرور تعاني منها المدن الرئيسية في لحظات الذروة.

Time	*Ms. Brown*	*Mr. Green*
T1	Reads the record Finds 25 seats left	
T2		Reads the record Finds 25 seats left
T3	Deducts 4 seats and writes updated record indicating 21 seats left	
T4		Deducts 6 seats and writes updated record indicating 19 seats left
	But at this point the record should show 15 seats left!	

شكل 12.2

مشكلة في سلامة البيانات بسبب التجديد المتزامن

Time	*Mr. White*	*Ms. Black*
T1	Gets and locks the record for nuts	
T2		Gets and locks the record for bolts
T3	Tries to get (and lock) the record for bolts and finds it locked by Ms. Black	
T4		Tries to get (and lock) the record for nuts but finds it locked by Mr. White
	DEADLOCK!	

شكل 12.3

وضع إغلاق الموت بسبب استخدام الإغلاق مع التجديد

لحل مشكلة إغلاق الموت يمكننا اعتبار اجراءات يمكنها أن تمنع من حدوثه أو اجراءات تكتشفه عندما يحدث وذلك لاتخاذ الاجراءات التصحيحية. وبينما يبدو المنع أفضل من الاكتشاف إلا أن المنع ليس سهلاً دائماً. وطبقاً لطبيعة الاستفسار فقد يكون على البرنامج أن يستخدم معلومات في قطعة بيانات واحدة والتي يجدها ويغلقها لتحديد ما هي قطعة البيانات الأخرى التي يجب أن يجدها ويغلقها. فإذا كانت هذه القطعة الثانية مغلقة فعلاً فيتوقف الاستفسار في منتصفه ممسكاً بقطعة بيانات هو في حاجة لها مغلقة مع عدم إتاحة القطعة الثانية. بالإضافة إلى ذلك فهناك مواضيع مرتبطة بالأداء تجعل من منع الإغلاق الميت غير مرغوباً فيه.

وعلى هذا فالثقة thrust المعتادة في هذه الحالة هي السماح بحدوث الإغلاق الميت واكتشافه عند حدوثه ثم علاجه. يمكن أن يكتشف الإغلاق الميت أثناء مرور الوقت أو من خلال عملية تتبع لسلسلة من الاستفسارات المنتظرة لمعرفة ما إذا كانت السلسلة تبدأ وتنتهي عند نفس الاستفسار أم لا. مثال ذلك إذا ما انتظر Smith شيئاً كان Jones قد أغلقه وانتظر Jones شيئاً كان Williams قد أغلقه وانتظر Williams شيئاً كان Smith قد أغلقه فيكون هناك إغلاق ميت. والحل هو اختيار أحد استفسارات الإغلاق الميت وعودته لأصله (أنظر القسم التالي الخاص بالاستعادة) دون عمل أي تغييرات حدثت على قاعدة البيانات عند هذه النقطة من التنفيذ. وبالطبع مع إزالة كل الإغلاقات يمكن لكل الاستفسارات الأخرى أن تستخدم الآن البيانات التي كانت مغلقة ثم يعاد بدء الاستفسار الذي أعيد إلى أصله بعد ذلك.

الاحتياطي والاستعادة : Backup and Recovery

علينا أن نفترض بغض النظر عما أصبحت عليه درجة تعقيد أجهزة الكمبيوتر أن طبقة مختلفة قد تحدث في وقت أو آخر على البيانات المخزنة. يمكن أن تأتي المشاكل من أشياء بسيطة مثل إدخال أحد المستفيدين قيم بيانات خاطئة داخل حقل أو من كارثة طبيعية مثل الحريق أو بعض الكوارث التي تدمر محتويات مركز الكمبيوتر وكل شيء موجود به. وعلى هذا يمكن أن تتراوح النتائج في ما يتبعها من حدوث خطأ في أحد قيم البيانات الفردية إلى تدمير كل ملفات أو قاعدة

بيانات المؤسسة مع وجود العديد من الإمكانيات بين هذين الحدين. وعلينا أن نفترض في أعمال تشغيل البيانات أنه سوف يحدث شيء خطأ من وقت لآخر بالنسبة للبيانات ويجب أن يكون لدينا الوسائل المتاحة لتصحيح ذلك وإعادة تشييده. ومن الواضح في بيئة قاعدة البيانات أن هذه الوسائل يجب أن تكون عامة بدرجة كافية لمعالجة الأنواع المختلفة من البيانات وأحجام البيانات المخزنة في قاعدة البيانات. وتأتي العمليات هذه تحت عنوان الاحتياطي والاستعادة Backup and Recovery.

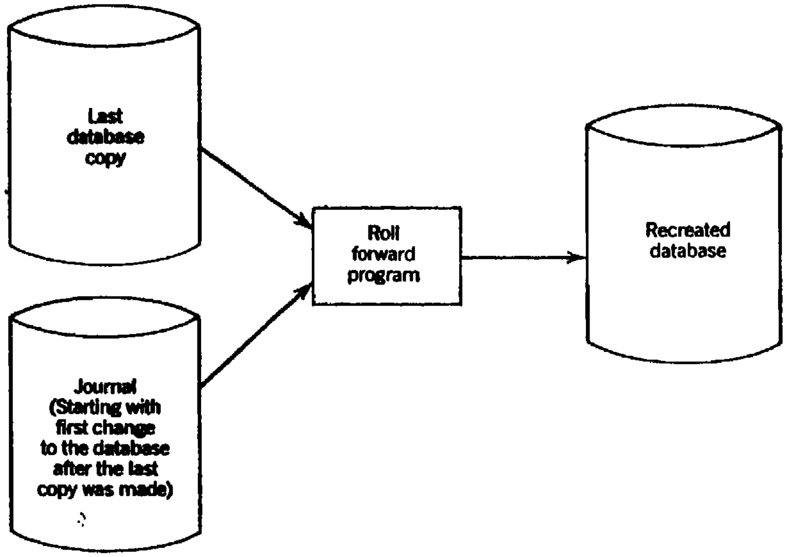
والأفكار الأساسية في الاحتياطي والاستعادة مباشرة جداً في مفهومها وقد استخدمت لوقت طويل. وهذا لا يعني شمول عدم وجود مكان للتغيير أو أنه لا يمكن لتنفيذ الأفكار أن يصبح معقداً. فلقاعدة البيانات التي يراد لها أن تكون قادرة على الاستعادة من أخطاء البيانات أو خلل في القرص أو ما شابه ذلك يجب اتخاذ اجرائين أساسيين أحدهما على أساس جدوله منتظمة والآخر على أساس نشاط ثابت.

وبالنسبة لأساس الجدولة المنتظمة يجب أن يكون هناك نسخة احتياطية من ملفات قاعدة البيانات، أي توجد نسخة من كل ملف من الملفات، النسخة (أو نسخة النسخة) يجب أن تحفظ في مبنى مختلف أو حتى في مدينة أخرى حتى إذا حدثت كارثة للكمبيوتر فلا تفقد كل النسخ. مثال ذلك الملفات الموضوعة على أقراص تنسخ على أقراص أخرى.

ومنذ هذه اللحظة وطبقاً لأساس ثابت يجب أن يسجل كل تغيير يحدث على قاعدة البيانات في ملف آخر يسمى سجل أداء log أو جريدة Journal. ويمكن حفظ الجريدة على شريط أو على قرص بالرغم من أن الاتجاه الحالي هو حفظها على قرص. وبينما يمكن لمحتويات الجريدة أن تتغير إلا أن الترميز الأساسي هو تسجيل قطعة البيانات التي حدث لها تغيير في عملية جارية معطاة وما كانت قيمتها قبل التغيير وماذا أصبحت قيمتها بعد التغيير. بعد ذلك يكون السؤال هو كيف يمكن استخدام نسخة قاعدة البيانات والجريدة لحل المشاكل المختلفة للاستعادة.

أولاً دعنا نعتبر حالة الكارثة التي تدمر مجموعة أقراص. تسمى عملية

الاستعادة في هذه الحالة إستعادة أمامية forward recovery أو الدوران للأمام roll forward شكل 12.4 . ولإعادة بناء القرص الذي فقد فإنك تبدأ بقراءة آخر نسخة أعدت للبيانات المفقودة والجديدة الخاصة بالتغييرات التي طرأت عليها. بعد ذلك يبدأ برنامج في قراءة محتويات الجريدة بدءاً من أول جزء تم تسجيله بعد إعداد نسخة قاعدة البيانات ويستمر خلال الجريدة حتى آخر جزء تم تسجيله قبل أن يدمر القرص مباشرة. وبالنسبة لكل من محتويات الجريدة هذه يغير البرنامج قيم البيانات المناظرة في نسخة قاعدة البيانات إلى القيمة المبنية في محتوى الجريدة. وهذا يعني أي تشغيل قد حدث في العملية الجارية التي تسببت في عمل محتوى الجريدة. والنتيجة النهائية هي إعادة تخزين قاعدة البيانات بعد إجراء التغييرات عليها. وأداء هذا العمل لكل عملية جارية تسببت في تغيير في قاعدة البيانات منذ إعداد نسخة لها (مثال القول لكل محتوى من محتويات الجريدة) بنفس الترتيب الذي نفذت فيه العمليات الجارية فعلاً سوف يجعل نسخة قاعدة البيانات في أحدث مستوى لها عند حدوث التدمير.



شكل 12.4

الدوران للأمام (استعادة أمامية)

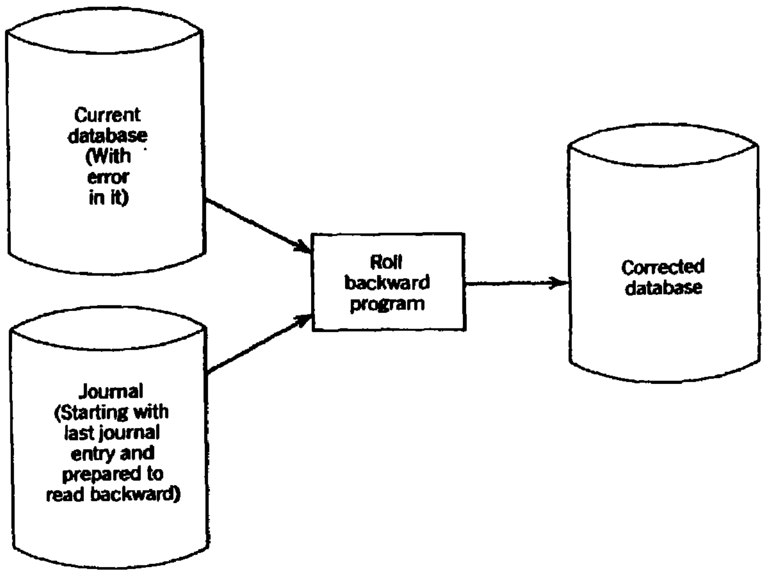
وهناك اختلافان على الاستعادة الأمامية يستحقان الذكر. أحد الاختلافات مبني على التمييز أن العديد من التغييرات قد تكون حدثت على نفس قطعة البيانات منذ إعداد آخر نسخة لقاعدة البيانات. إذا كان هذا هو الحال فإن آخر هذه التغييرات فقط التي تبين بعد كل شيء فيه قطعة البيانات هذه عند النقطة التي تدمر فيها القرص تحتاج لأن تستخدم في تجديد نسخة قاعدة البيانات في عملية الدوران للأمام. فإذا كانت بيئة قاعدة البيانات قابلة للتقلب ويحدث فيها تغييرات بصورة متكررة، ومن المعتاد بنفس قطعة البيانات أن تتجدد العديد من المرات بين عمليات نسخ قاعدة البيانات، فعند ذلك يكون الدوران للأمام كما سبق وصفه غير كفوء. بدلاً من ذلك قد يكون من المفيد ترتيب الجريدة قبل إجراء عملية الدوران للأمام وذلك لإيجاد آخر تغيير حدث لكل قطعة بيانات سبق تجديدها منذ إعداد آخر نسخة لقاعدة البيانات. وعند ذلك فإن آخر تغييرات فقط تحتاج إلى إدخالها على نسخة قاعدة البيانات في عملية الدوران للأمام.

وهناك اختلاف آخر للدوران للأمام وهو تسجيل تحديد لما تشبهه العملية الجارية عند نقطة التنفيذ، بدلاً من التسجيل قبل وبعد صور البيانات في الجريدة، مع المعلومات المفيدة الأخرى. وعند ذلك تصبح عملية الدوران للأمام إحدى عمليات إعادة تنفيذ كل العمليات الجارية بترتيب تنفيذها الأصلي. ويأخذ هذا وقتاً أطول كعملية استعادة.

والآن دعنا نعتبر موقفاً مختلفاً. إفرض أن منتصف العملية المعتادة اكتشف أحد الأخطاء الذي يشمل قطعة بيانات مجددة حديثاً. ويمكن أن يكون السبب بسيطاً مثل خطأ آدمي في كتابه إحدى القيم أو معقداً مثل انتهاء برنامج نهاية غير طبيعية وترك بعض، وليس كل، التغييرات التي حدثت على قاعدة البيانات التي يفترض أنها حدثت. لماذا لا تصحح هذه البيانات غير الصحيحة مع عدم عمل شيء كبير من ذلك؟ لأنه في فترة معينة يمكن لبرامج أخرى أن تقرأ البيانات غير الصحيحة وتستخدمها مكونة خطأ في أماكن أخرى في قاعدة البيانات.

وعلى هذا فالخطأ المكتشف وفي الواقع كل التغييرات الأخرى التي حدثت على قاعدة البيانات منذ اكتشاف الخطأ يجب أن تكون أخرجت backed-out. وتسمى هذه العملية بالاستعادة الخلفية backward recovery أو بالدوران الخلفي

roll-backward شكل 12.5. والفكرة هي بالضرورة البدء بقاعدة البيانات في حالتها الحالية (ملاحظة: نسخ قاعدة البيانات لا تفعل شيء مع هذا الوضع). والجريدة بدءاً بآخر محتوى أدخل فيها. وبعد ذلك يقرأ البرنامج للخلف backward من الجريدة مع تصحيح كل قيمة بيانات محددة إلى ما قبل صورتها before image كما هي مسجلة في الجريدة حتى يصل إلى النقطة التي حدث فيها خطأ. ولا يعمل undoes البرنامج كل عملية جارية في الترتيب العكسي (الأخير أولاً). الذي أعد منه. فإذا أثر الخطأ في الواقع على عمليات جارية أخرى فيجب إعادة تشغيل هذه العمليات الجارية ويمكن أن تحدث هذه العملية يدوياً أو آلياً طبقاً لعدد العوامل بما في ذلك طبيعة المعلومات المحفوظة في الجريدة.



شكل 12.5

الدوران للخلف (استعادة خلفية)

وهناك ملاحظتان على الاستعادة. بعض النظم قادرة على بدء عملية الدوران للخلف تلقائياً لعدم عمل undo التغييرات التي أجريت على قاعدة البيانات بإتمام جزئي ثم توقفت العمليات الجارية أو جعلتها تتم بصورة خاطئة. ويسمى ذلك بالطرد الديناميكي dynamic backout. وهناك مواقف يكون من المفيد فيها إعادة تخزين قاعدة البيانات عند نقطة يكون عندها ثقة بأن كل التغييرات التي حدثت على قاعدة البيانات حتى هذه النقطة دقيقة. وبعض النظم

قادرة على كتابة سجل خاص للجريدة يعرف بأنه نقطة التأكد checkpoint يحدد نوع حالة التوازن.

القابلية لإجراء المراجعة المحاسبية : Audiability

كموضوع حظي باهتمام بسيط نسبياً في الماضي لكنه يكتسب أهمية متزايدة مع الوقت نجد موضوع القابلية لإجراء المراجعة المحاسبية لقاعدة البيانات. ويمكن أن يكون لاصطلاح القابلية للمراجعة المحاسبية معاني Connotaions عديدة وسوف نمسه بالنسبة إلى زوج منها فقط.

أحد أوجه قابلية قاعدة البيانات للمراجعة المحاسبية هو حصر بيئة قاعدة البيانات للتأكد أن البيانات يحدث لها حماية مناسبة. وهذا يعني وجود اجراءات أمن كافية وإجراءات احتياطية واستعادة وتزامن كافية. وطبقاً لعمق المراجعة المحاسبية فقد يعني ذلك أن نظام إدارة قاعدة البيانات يستخدم كذلك بصورة مناسبة وأنه هناك تمتع بمزاياه على مستوى مقبول.

وهناك وجه آخر لقابلية قاعدة البيانات للمراجعة المحاسبية وهو المقدرة على التتبع الخلفي لمعرفة من الذي قام بالتداخل مع قاعدة البيانات. ويمكن أن يكون هذا على مستوى من فعل أي التغييرات في قاعدة البيانات أو يمكن أن يكون على مستوى مبسط لمن استرجع أي البيانات من قاعدة البيانات. وأوضح وسيلة لتلبية هذا المتطلب هو الجريدة. في الواقع يمكن استخدام نفس الجريدة التي تستخدم في عمليات الاستعادة وذلك في أغراض المراجعة المحاسبية. كذلك إذا كان الهدف من تقديم محاولة مراجعة «audit trial» هو المقدرة على حفظ تتبع من الذي أجرى تغييراً في قاعدة البيانات، فيجب أن تحتوي الجريدة على من الذي أجرى التعديلات المحددة في عناصر الجريدة. وإذا كانت محاولة المراجعة بغرض حفظ تتبع استرجاعات بسيطة من قاعدة البيانات أو تجديدات لها، فعلى الجريدة أن تشمل محتويات لقراءات بسيطة بالإضافة إلى محتوياتها المعتادة للتجديد. وعلى أية حال عليك أن تلاحظ أن آخر متطلب في النظم ذات معدلات النشاط المرتفعة سوف ينتج عنه جرائد ضخمة وسوف يبطىء ذلك من عمل النظام الكلي مع تسجيل كل اتصال بقاعدة البيانات في الجريدة.

مراجع: References

Date, C.J., Introduction to Database Systems, vol. 2, Addison - Wesley, Reading, MA, 1983.

Denning, D.E., and Denning, P.J., «Data Security», Computing Surveys, vol. 11, no. 3, September, 1979, pp. 227 - 250.

Fernandez, E.B., Summers, R.C., and Wood, C., Database Security and Integrity, Addison - Wesley, Reading, MA, 1981.

Hsiao, D.K., Kerr, D.S., and Madnick, S.E., Computer Security, Academic Press, New York, 1979.

King, J.M., Evaluating Data Base Management Systems, Van Nostrand Reinhold, New York, 1981.

Kroenke, D., Database Processing, 2nd ed., Science Research Associates, Chicago, IL, 1983.

Leiss, E.L., Principles of Data Security, Plenum Press, New York, 1982.

Perry, W.E., Ensuring Data Base Integrity, Wiley, New York, 1982.

Ross, R.G., Data Base Systems, AMACOM, New York, 1978.

Verhofstad, J.S.M., «Recovery Techniques for Database Systems», Computing Surveys, vol. 10, no. 2, June 1978, pp. 167 - 195.

أسئلة وتمارين: Questions and Exercises

١ ـ صف بكلمات من عندك أنواع تهديدات الأمن المختلفة لبيانات الشركة. هل تعتقد أن التهديدات الداخلية أو الخارجية هي الأكثر خطورة؟ لماذا؟

٢ ـ صف بكلمات من عندك أنواع وحدات حماية الأمن المختلفة التي يمكن استخدامها في حماية بيانات الشركة. هل أحد هذه الأنواع يكون أكثر أهمية عن الآخرين؟ لماذا؟

٣ ـ إعمل قائمة بأنواع التهديدات المختلفة ووحدات حماية الأمن المختلفة وحدد كل وحدة من وحدات الحماية تحمي ضد أي نوع من أنواع التهديدات.

٤ ـ ما تأثيرات الأمن لجهاز كمبيوتر ينتج أرقاماً عشوائية ككلمات مرور (أي أن كلمات المرور تصبح صعبة للشخص أن يتذكرها)؟

٥ - ما تأثيرات الأمن الخاصة باستخدام بصمات الأصابع أو التوقيع أو الصوت بدلاً من كلمات المرور وذلك لتمييز النظام للشخص؟ ماذا يحدث إذا ما كانت وحدة التمييز ليست تامة الدقة دائماً؟

٦ - إعمل طريقة بسيطة لخلط البيانات. ما الوقت اللازم لكي يستطيع أحد الأفراد فك هذا الخلط؟

٧ - إعمل عينة لموقف توضح مشكلة التجديد المتزامن مع الإغلاق.

٨ - إعمل عينة لموقف توضح مشكلة الإغلاق الميت.

٩ - هل يمكنك أن تتصور موقفاً يجب فيه على الشخص الذي يقوم بعمل استرجاع بسيط أن يبعد حتى يتم إجراء عملية تجديد؟

١٠ - إذا طلبت استفسارات مختلفة تحتاج سجلات متعددة من نفس الملف المحمي بالإغلاق وتطلب هذه السجلات بصورة دائمة في ترتيب تصاعدي طبقاً لقيمة حقل فردي، هل يمكن للإغلاق الميت أن يحدث في أي وقت من الأوقات؟ لماذا أو لم لا؟

١١ -صف بكلمات من عندك الأنواع المختلفة للتهديدات المدمرة لبيانات الشركة.

١٢ -من الأنواع المختلفة للتهديدات المدمرة لبيانات الشركة أي منها يتطلب استعادة أمامية وأيها يتطلب استعادة خلفية؟

١٣ -اعتبر العناصر المختلفة التي يمكن أن تكتب في جريدة لعملية تجديد بما في ذلك صورة البيانات قبل التغيير وصورتها بعد التغيير وكذلك وصف العملية الجارية نفسها. ماذا يكون مطلوباً للاستعادة الأمامية، وماذا يكون مطلوباً للاستعادة الخلفية؟ وضح إجابتك.

١٤ - ما العوامل التي تعتقد أنها تحكم تكرار عمل نسخ من قاعدة البيانات بالنسبة لعمليات الاستعادة؟ ما مميزات وعيوب عمل نسخ بصورة متكررة أكبر أو أقل لقاعدة البيانات؟

١٥ -ناقش أسباب وجود المراجعة المحاسبية للبيانات في الجريدة هل هناك عيوب في عمل ذلك؟.

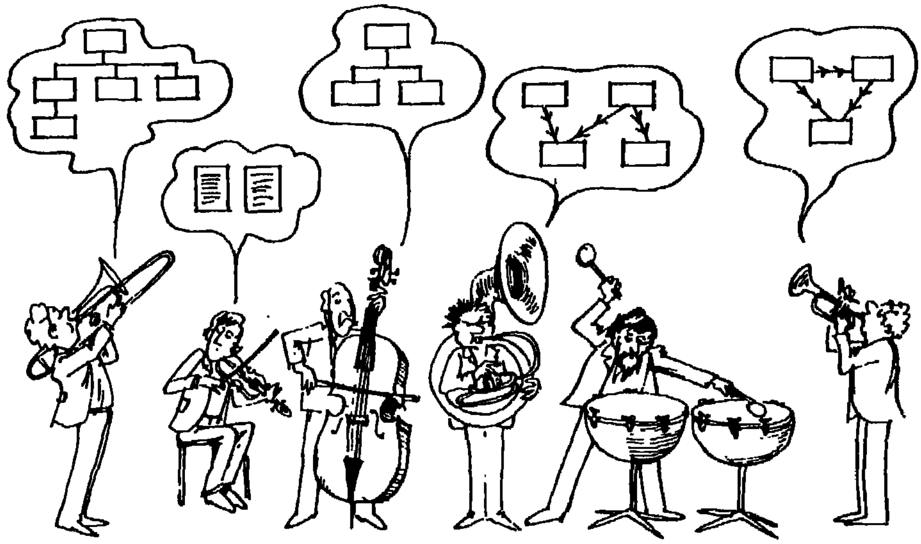
الفصل الثالث عشر
إدارة البيانات
Data Administration

مقدمة : Introduction

حتى الآن في هذا الكتاب كان اهتمامنا أساساً منصباً على الأوجه التقنية لقاعدة البيانات وبيئة قاعدة البيانات. ولكن كما هو الحال مع مكونات أي تقنية فيجب أن توضع للعمل أي ليعمل بها الناس وإلا فإنها لا تزيد عن كونها نظرية. بالطبع هناك العديد من الأشخاص في أوساط تشغيل البيانات الموجهة ناحية قاعدة البيانات يقومون بأداء أعمالهم الفردية. ويكتب المبرمجون برامج لتطبيقات محددة باستدعاءات نظم إدارة قواعد البيانات محتواه داخلها. ويصمم المصممون قواعد البيانات للتطبيقات الفردية. ويبحث المستفيدون النهائيون عن معلومات باستخدام لغات الاستفسار. ويبني مبرمجو النظم مجموعات تحكم لنظم إدارة قواعد البيانات.

إلا أن كل هؤلاء الناس يؤدون أعمالهم الفردية للتطبيقات الفردية وينتج عن ذلك بيئة مشوشة دون وجود أي نوع من أنواع التنسيق. وبعض هذه التشويشات تظهر على أنها تكرارات للبيانات وبعضها يمكن أن يتسبب في مشاكل لأداء وحدة التشغيل المركزية التي يشترك العديد من المستفيدين في استخدامها. وقد يبين بعضها الآخر مشاكل في صيانة نظم البرامج وذلك بسبب الإفتقار إلى النمطية في تصميم قاعدة البيانات وفي اعتبارات البرنامج. والمشكلة ليست عملية عدم المقدرة على تنفيذ جزء من عمل بعض العاملين في تشغيل البيانات ولكنها مشكلة الافتقار إلى توجيه شامل وتحكم شامل. وهي تشبه بعض الشيء عمل الأوركسترا بدون قائد. فكل موسيقي يمكن أن يكون موهوباً في حد ذاته ولكن بدون وجود

شخص يبرمج أعمال الموسيقيين مع بعضهم فلا تكون النتيجة أقل من إزعاج كما هو موضح في شكل 13.1 .



شكل 13.1

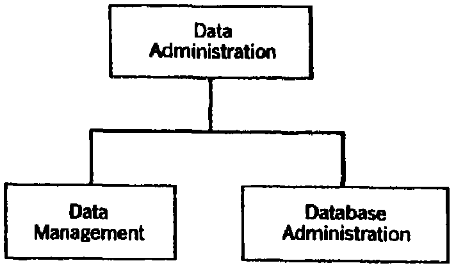
إدارة البيانات تقوم بالتنسيق مثلما يفعل المايسترو في الأوركسترا

في نهاية الستينيات الميلادية ومع بدء استخدام نظم إدارة قواعد البيانات بدأت قلة من الشركات التي تنظر إلى الأمام في تمييز الحاجة لمجموعة مستقلة يكون عملها عبارة عن إدارة بيئة قاعدة البيانات. ومع مرور السنوات اكتسبت بعض هذه المجموعات مسئولية عن البيانات غير الموجودة في ملفات نظام إدارة قاعدة البيانات كذلك. بالإضافة إلى ذلك نضجت بعض المجموعات من موقع إدارة البيانات فقط على مستوى العمليات لأداء التخطيط الاستراتيجي بجانب ذلك وكذلك أداء أنشطة أوسع.

وسوف نستكشف في هذا الفصل هذا الحقل الجزئي من حقول قاعدة البيانات والذي يسمى بصفة عامة الآن بإدارة البيانات data administration. وسوف نفحص مميزات إدارة البيانات بواسطة مجموعة إدارة البيانات والمسئوليات التي تقع على كاهل هذه المجموعة وفئات العمل التي تقوم بها المجموعة وموقع

المجمـوعة من تنـظيم تشغيل البيـانات وتـأثير إدارة البيـانات عـلى تنـظيم تشغيـل البيانات واتجاهات المستقبل.

لقد طورت الاصطلاحات التي سوف ترد فيـها بعد بـواسطة شركـة Guide International Inc. ويشير اصطلاح قسم إدارة البيـانات data administration إلى مجموعة مسئولة عن إدارة بيانات المؤسسة كأحـد موارد المؤسسـة الهامـة. وهي مسئولية تطوير وإدارة السياسات والاجراءات والخطط والعمليات لتعريف وتنـظيم وحماية واستخدام البيانات بكفاءة داخل المؤسسة. وكما يوضح شكل 13.2 تتكـون إدارة البيـانـات من جـزئـين يمكن أن يكـونـا قسمـين مستقلين إذا كـانت المؤسسـة كبيرة بدرجة كافية وذلك تحت مـظلة إدارة البيانـات. الوظيفتـان الجزئيتـان هما إدارة البيانات data management وإدارة قاعدة البيانات database administration. ومجموعة إدارة البيانات تقوم بالتخـطيط والتحليل وتكـون مسئولـة عن السياسـات ووضـع النمطيـات والتعليم ومراقبـة موارد البيـانات وعمـل المحاسبـات الخاصـة بالبيانات وتقدم الدعم لمحلل النظم والمبرمجين أثناء تطوير التطبيقات. أما مجمـوعة إدارة قاعدة البيانات فتكون موجهة الى العمليات بصورة أكبر حيث تكون مسئولـة عن التوجيه اليومي لقاعدة البيانات (وفي بعض الحالات لبيانات أخرى كذلـك). وتنفذ مجموعة إدارة قاعدة البيانات السياسات التي تضعها مجمـوعة إدارة البيـانات وذلك على مستوى العمليات. وسوف نستكشـف مدى كامل للمسئـوليات بعملهـا أكثر فيها بعد. لكن لاحظ أنه حتى عند هذه النقـطة المبكرة لا تـوجد قـاعدة ثـابتة عن فئة مسئوليـات مجموعـة إدارة البيانـات data administration أو عما يجب أن تؤديه أجزاؤها لتعتبر أنها ناجحة أو أنها عارية تماماً.



شكل 13.2

مميزات منهج إدارة البيانات :

Advantages of Data Administration Approach

سبق أن بينـا مميزات منهـج قاعـدة البيانـات بالنسبـة لتشغيـل البيـانـات في الفصـل الرابـع من الكتاب. وسـوف نوضـح في هذا القسم مميـزات منهـج إدارة البيانات لإدارة البيانات سواء كان ذلك في جزء قاعدة البيانات أو في أي جزء آخر غير قاعدة البيانات من بيئة تشغيل البيانات.

البيانات كمورد مشترك للمؤسسة :

كما سبق ذكره يجب أن تعتبر البيانات كمورد هام للمؤسسة مثل المورد المـالي أو المبـاني والمعدات أو المـورد البشري وما إلى ذلـك. وقد اقـترح بعض الناس أن البيـانات هي أكـثر الموارد أهميـة للمؤسسة لأنها بسبب طبيعتهـا تصف كل المـوارد الأخرى. ووجود قواعد بيـانات منـظمة يمكن الاتصـال بها يميـل إلى أن يزيـد من كمية اجراءات أعمال المؤسسة التي تستخدم تشغيل البيانات وتقـود إلى كفاءة أكـثر بصفة عامة للمؤسسة. والاستخدام على مـدى واسع للبيـانات وتشغيـل البيانـات يمكن أن يوسع من موقف المؤسسة التنافسي بجعله أكثر استجابة في أعـين عملائهـا وموردبها.

كما أن البيانات مورد مشترك كذلك. وأحد العقائد الأسـاسية لمنهج قاعـدة البيانات هو تقليل تكرار البيانات بهدف تحسين سلامتهـا وتقليل تكـاليفها. وأحـد طرق تقليل تكرار البيانات هي وجود مستفيدين مختلفين وتطبيقات مختلفـة يكونـوا في حـاجة إلى المشـاركة في بعض البيـانات في صـورة ملف واحد بـدلاً من وجـود نسخة خاصـة لكل مستفيـد أو كل تـطبيق في ملف واقعي مختلف. وبمصطلحـات البيانات فأحد صور المشاركة الأخرى هي المشاركة في نظم مكونات الكمبيوتر التي تقوم بتشغيل البيانات ويمكن للضغط عـلى موارد نـظم المكونـات هذه التي يحـدثها أحد التطبيقات أن يؤثر بطريقة متكررة على أداء التطبيقات الأخرى.

ما يؤدي إليه كل ذلك هو ببساطة ما يلي: أي مورد مشترك للمؤسسة بمـا في ذلك البيانات يجب أن يديـرها مجمـوعة من العـاملين في المؤسسة والتي نسميهـا في حالة البيانات بقسم إدارة البيانات. فلا يكون هناك معنى لوجـود مورد هـام لا يدار

على الإطلاق أو يدار جزئياً بواسطة مجموعة لديها مسئوليات أخرى. كما أنه لا يكون هناك معنى لوجود أي مجموعة من المجموعات تتنافس على الموارد المشتركة (وليكن قسم تطوير تطبيقات في هذه الحالة) ويديرها كذلك ـ فيجب أن يكون مدير المورد حيادياً عندما يحدث أي نزاع.

الكفاءة في التخصص في العمل:

هناك العديد من الوظائف التي تشملها إدارة البيانات data management والتي تتراوح من التخطيط طويل المدى ووضع النمطيات لتصميم قاعدة البيانات وتوجيه أداء قاعدة البيانات. (سوف نعد قائمة موجزة بذلك). وبعض هذه الوظائف مثل تصميم قاعدة البيانات على سبيل المثال يجب أن ينفذه شخص معين موجود في تنظيم تشغيل البيانات وإلا فلن يحدث أي تشغيل للبيانات. وعلى سبيل المثال فالوظائف الأخرى مثل التخطيط على المدى الطويل تكون اختيارية بعض الشيء طبقاً لدرجة تعقيد بيئة تشغيل البيانات. إلا أنه هناك شيء أكيد عن كل الوظائف وهو أنها كلها تتطلب درجة مرتفعة من تخصص العاملين للحصول على نتائج مثالية.

عند ذلك يصبح السؤال خاصاً بتحديد الشخص المسئول عن أداء هذه الأنشطة من العاملين في تشغيل البيانات. وهناك امكانيتان أساسيتان. إحداهما هي تجزئة الأنشطة على عدة عاملين مختلفين في تشغيل البيانات تكون مسؤولياتهم العادية متقاربة مع الأنشطة اللازمة لإدارة البيانات. ويقوم المبرمجون الجيدون بتصميم قاعدة البيانات، ويعد مبرمجو النظم توجيهات عمل قاعدة البيانات وهكذا. الإمكانية الثانية تكون عمل قسم مستقل لإدارة البيانات وذلك لتحمل هذه المسئولية.

والاختيار الأخير هو الأفضل نظراً لتعقيد الأنشطة ونظراً لأهمية تنفيذ مكونات الأنشطة بالنسبة للتشغيل الكفوء لبيئة تشغيل البيانات. وهناك ببساطة الكثير من معدي التطبيقات ومن مبرمجي النظم والعاملين الآخرين في تشغيل البيانات الذين يؤدون الكثير من الأعمال الأخرى بالفعل. ويتوقع لكل منهم أن يصبح ملماً بأداء أنشطة محددة للبيانات مثل الأشخاص الذين يصبحون متخصصين في إدارة البيانات ولا يؤدون سوى أنشطة إدارة البيانات يوماً بعد يوم.

والحجة نمطية جداً بـالنسبة للتخصص في العمـل بصفة عـامة وهي مهمـة في هذا الموقف بصفة خاصة مثل أي موقف آخر.

تفاصيل البيانات مأخوذة من المبرمجين:

بـالإشارة إلى القسم السـابق ومع الأهميـة الكافيـة لمعالجـة السؤال الخـاص بـالعلاقـة بين الأفـراد العاملين في إدارة البيـانات والمـبرمجين فهـذا يجعل المـوضوع يستحق المعالجة المنفصلة. فمع ظهور منتجات التطبيقـات (لغات الجيـل الرابـع) ووسائل المراوغة الأخرى والوسائل الأسرع من عملية البرمجة فإن الـطبيعة الـدقيقة لعمل المبرمجين أصبحت موضـع جدال. وهنـاك كذلـك وجه لهـذا الجدال مـرتبط بالبيانات.

أحد الأنشطة التقليـدية للبرمجـة هي كتابـة شفرة تصف تكـوين البيانـات. وربما يكون أفضـل مثال هـو جزء التعـريف من برنـامج الكـوبل. لكن من المُعَدّ بصورة أفضل لكتابة جزء التعريف من بـرنامـج الكوبـل أو ما يكـافئه في اللغـات الأخرى هل هو مبرمج التطبيقات أم إداري البيانات؟ القول هنـا هو أن هـذه هي منطقة أخرى للبيانات المكثفة التي يصبح إداري البيانـات متخصصاً فيهـا ويسمح للمبرمج أن يركز على منطق البرنامـج والذي يجب أن يكـون المنطق ذات الاهتـمام الأكبر له في الأساس. هنا ومرة أخرى يجب أن ينتج عامل التخصص أخطاء أقـل في شفرة وصف البيانات.

إدارة العمليات للبيانات:

لقد سبق أن نظرنا الى البيانات على أنها مـورد من مستوى مـرتفع. وهنـاك وجهة نظر أخـرى للبيانـات كمورد وهي رؤيتهـا من مستوى عمليـات منخفض. فقسم إدارة البيـانات ضروري لـلإدارة والتحكم المناسبـين للبيانـات على مستـوى العمليات.

وللبـداية لا يـوجد اختيـار ثان بعـد إدارة البيانـات كمكان لـوضع مـراقبـة التشغيـل لقامـوس البيانـات. فإدارة البيـانات مسئولة عن إدارة بيـانات الشركـة وقاموس البيانات هو مخزن للمعلومات عن البيانات الموجـودة ومكانها ومن يتصـل بها وما إلى ذلك. بالإضافة إلى ذلك فإن قسم إدارة البيانات المتداخل مـع الأقسام

الأخرى يكون في أحسن حالاته لرؤية أن قاموس البيانات مستمر في أنه في أفضل أهم وأدق مورد معلومات لكل العاملين المختلفين في تشغيل البيانات الذين يشيرون إليه كذلك.

وفي بيئة البيانات المشتركة يكون هناك دائماً بعض التطبيقات أو بعض المستفيدين الذين يعتمدون على تطبيقات أو مستفيدين آخرين في جمع البيانات أو في تحديث الملفات بصفة دائمة أو غير دائمة. ومن الواضح أنه من الأفضل أن توجد مجموعة مستقلة لإدارة البيانات تحتفظ بتتبع من المسئول عن تجديد أي الملفات وتوصيهم بالحفاظ على الجدولة المتوقعة وذلك لمنفعة كل الآخرين الذين يستخدمون هذه الملفات.

وهناك عدد من الأنشطة الأخرى المرتبطة بالبيانات والتي يجب أن تدار بواسطة المتخصصين في إدارة البيانات وذلك لمنفعة كل المعتمدين على البيانات. فأمن البيانات أي منع إظهار بيانات الشركة للعالم الخارجي وخصوصية البيانات أي حفظ البيانات الحساسة عن العملاء والموردين والعاملين مقيدة هما موضوعين من هذه المواضيع. كما يوجد كذلك السؤال الخاص بإدارة (وليس بالضرورة تنفيذ كل التفاصيل) النسخ الاحتياطية والاسترجاع عندما يحدث ضرر في قاعدة البيانات. كما يوجد كذلك الأعمال المرتبطة باتصالات البيانات وإذاعة البيانات: حفظ سجل بتفاصيل بيئة البيانات (أي من خلال قاموس البيانات) وجعل بقية المؤسسة تعرف أي البيانات متاحة للاستخدام في بيئة البيانات المشتركة.

مسؤوليات إدارة البيانات:

Data Administration Responsibility

كما سبق القول فلا يوجد شيء صحيح وشيء خطأ في مجموعة الأنشطة التي تميز قسم إدارة البيانات. وبالمثل فهناك بعض المرونة في توزيع الأنشطة على إدارة البيانات data management وإدارة قاعدة البيانات database administration. ومجموعة الأنشطة المحددة لإدارة البيانات وتقسيمها بين إدارة بيانات data management وإدارة قاعدة بيانات تعتمد على نضج المنظمة وعلى الجو التقني والسياسي لمنظمة تشغيل البيانات وعلى الشركة ككل.

وسوف نستكشف في هذا القسم مدى واسعاً لمسئوليات إدارة البيانات يتبـع بصفة عامة تجزئة GUIDE الدولية بين إدارة البيانات وإدارة قاعدة البيانات. تـذكر أن إدارة البيــانــات هي وجــه التخــطيط والتـحليــل لإدارة الـبيــانــات data administration بينما إدارة قاعدة البيانات هي وجه العمليات.

مسئوليات إدارة management البيانات:

تخطيط البيانـات: مع معـاملة البيانـات كمورد هـام مشترك يجب أن تؤخـذ الاحتياطات في التخطيط لمستقبلها.

وأهم وجه للتخطيط في هذا الصدد هو تحديد أي البيانات تكون مطلوبة في أعـداد تطبيقـات المستقبل وكيفيـة تكامـل البيانـات مع مكـوناتهـا ومـع البيـانـات الموجودة. وهناك عدد من الـطرق التي أعدت لتـطوير تخطيط البيانـات هذا. أحـد هذه الأمثلة هو BSP من شكرة IBM والتي سبق مناقشتها. مثل هذه الطرق تأخذ في الاعتبار عمليات الأعمال التي تؤديها الشركـة كجزء من عمليـاتها المعتـادة وتحدد أولويات لاستخدام الكمبيوتر معها في المستقبل. وبينما تعمل كلها بصفة عامة على مستـوى استراتيجي مـرتفع وقـد لا تدخـل في تفاصيـل الحقول الفـرديـة وتصميم قاعدة البيانات إلا أنها تقدم خريطة طرق عريضة للعمل من خلالها.

وبالنسبة للتخطيط الاستراتيجي للبيانات فهو خاص بنظم المكونـات ونظم البرامج المطلوبة لدعـم تشغيل بيانات الشركة في المستقبل. فعادة ما يوجد في تنظيمات تشغيـل البيانـات الكبيرة أنـاس يعملون وقتـاً كـامـلاً في هـذا الشـأن ولكنهم من الـواضح أنهم يجب أن يـأخـذوا مـدخـلات من العـاملين في إدارة management البيـانات خـاصة بحجم البيـانات وبنـوع التشغيل الـذي يجري عليهـا. وتـتراوح الأسئلة المشمولة من مواضيع مبـاشرة نسبياً مثـل عدد مجمـوعات الأقـراص اللازم لاحتواء البيانـات إلى مواضيـع أوسع مثـل مدى قـدرة الحسابـات الأولية الـلازمة لدعم بيئة تشغيل البيانات الإجمالية.

وهناك صورة ثالثة لتخطيط إدارة management البيانات وهي كيف يـوضع قاموس البيانات معـد للاستخـدام. ويشمل ذلـك البيانـات التي يجب تخزينهـا في القاموس والاستخدامات التي يوضع القاموس معداً لها ومن يجب أن يتـداخل مـع

القاموس وكيفية التداخل وعلى أي أنواع الجدولة يجب أن توضع كل هذه الأمور بالنسبة للزمن.

ومع بناء الشركة للمزيد والمزيد من تشغيلها للبيانات حول مفهوم قاعدة البيانات فإنها تظل تواجه عدداً كبيراً من التطبيقات الموجودة التي وجدت قبل استخدام قاعدة البيانات. وعلى هذا فهناك وجه آخر لتخطيط إدارة البيانات والذي يشمل إدخال هذه التطبيقات داخل بيئة قاعدة البيانات: أي التطبيقات يجب إدخالها وطبقاً لأي جدولة وكيف يمكن توفير البيانات اللازمة لها في قاعدة البيانات؟

نمطيات قاعدة البيانات: تهدف إلى تقليل الأخطاء وتحسين الأداء واتساع مقدرة العاملين في تشغيل البيانات على فهم العمل الذي يؤديه العاملون الآخرون في تشغيل البيانات. وبصفة عامة لتقديم درجة مرتفعة من التنسيق في بيئة تشغيل البيانات فإن ذلك يكون اجبارياً على وظيفة إدارة البيانات ومقدرتها على وضع نمطيات خاصة بالبيانات واستخدمها. وأحد أمثلة النمطيات هو مراقبة طريقة تكوين أسماء الحقول وأسماء الملفات وأسماء البيانات الأخرى المرتبطة. وعادة ما يجب أن تغطي معنى كبيراً في رموز بسيطة بقدر الإمكان وكذلك يتفق عليها العاملون في إدارة البيانات. هناك مثال آخر وهو التركيز على التماسك في طريقة كتابة البرامج للاتصال بقاعدة البيانات وبصفة خاصة بالنسبة إلى الطريقة التي تكتب بها التعليمات التي تنادي على قاعدة البيانات. ويمكن للحذر هنا أن يساعد في منع نداء قاعدة البيانات من إحداث مشاكل في التنفيذ وكذلك تسهيل الصيانة عن طريق وجود نمطيات وتعليمات مفهومة بالفعل.

العلاقة المتبادلة لمحللي النظم مع المبرمجين: مثل العلاقة المتبادلة لمطوري البرامج مع العاملين في إدارة البيانات (والذين يسمون في العادة بمحللي نظم في هذا الدور) فإنه يخدم وظيفة أساسية. فهم مسئولون عن تقديم دعم لمحللي النظم والمبرمجين في كل الأمور الخاصة بالبيانات اللازمة للتطبيق. أثناء مرحلة تحليل النظام مع تطوير التطبيق فإن هذا الدعم يمكن أن يشمل المساعدة في تحديد البيانات اللازمة للتطبيق كما يشمل كذلك المساعدة في إيجاد أي عناصر البيانات ضروري للتطبيق الموجود بالفعل في قاعدة بيانات نشطة. وهناك صورة أخرى

للمساعدة لمطوري التطبيقات وهي المساعدة في إعداد برامج وصف البيانات مثل مواصفات جزء بيانات برنامج الكوبل.

هناك وجه آخر لنشاط العلاقة المتبادلة هذا، وهو موضوع مستقل بذاته، وهو السؤال الخاص بتصميم قاعدة البيانات. ليس هناك شك أن محللي البيانات يجب أن يكونوا موجودين في تصميم قاعدة البيانات على مستوى معين إلا أن القرار الخاص بهذا المستوى لاشتراكهم يجب أن يعتمد على عدد من العوامل. ففي بيئة تشغيل البيانات التي تكون فيها منظمة تشغيل البيانات قوية جداً ولا يوجد فيها كمية بيانات معنوية مشتركة بين التطبيقات المختلفة ومناطق الوظائف المختلفة للشركة يمكن أن يقوم محللوا البيانات بكل عمل لتصميم قاعدة البيانات بأنفسهم. وهنا مرة أخرى يمكنهم أن يقفوا كمجموعة مهمة تنتج أفضل تصميم للمنفعة العامة لكل المستفيدين. والاختيار الآخر هو مطوري التطبيقات وتكليفهم بعمل تصميم قاعدة البيانات وذلك بالاستشارة النشطة مع محللي البيانات أو بمسئولية الموافقة بعد أداء محللي البيانات للعمل. وفي دور الاستشارة النشطة يوجه محللوا البيانات خبراتهم لهذا المجهود وكذلك لتحديد كيف يجب أن تتداخل البيانات الجديدة مع البيانات الموجودة فعلاً في قاعدة البيانات إذا كان هناك مثل هذا الدمج. وفي دور الموافقة فإن مطوري التطبيقات (عادة ما يكون المبرمجون الجيدون في هذا النشاط) يصممون قاعدة البيانات. ولكن حيث أن العاملين في إدارة البيانات يكونوا مسئولين عن أداء قاعدة البيانات فيجب أن يكون لديهم مقدرة الموافقة (أو الرفض) حيث أن التصميم الضعيف لقاعدة البيانات لا يمكن أن يؤثر بصورة معادة فقط على أداء التطبيقات المستخدمة لهذه القاعدة لكنه يمكنه أن يبطىء كذلك من أداء التطبيقات الأخرى والتي تستخدم قواعد بيانات أخرى يتم تشغيلها على نفس وحدة التشغيل المركزية.

التدريب: العاملون في إدارة البيانات data management مسئولون عن تدريب كل العاملين في الشركة والذين عليهم أن يفهموا مفهوم قاعدة البيانات على العديد من المستويات. فأفراد الإدارة عليهم أن يعرفوا أسباب جودة قاعدة البيانات للشركة ولوظائفهم الفردية بصفة خاصة. ويجب أن يفهم المستفيدون لماذا يستخدم الآن مستفيدون آخرون ملفات بياناتهم المنفصلة السابقة ولماذا لاتزال

البيانات آمنة ومحتفظة بخصوصيتها. وبالطبع يجب تدريب مطوري التطبيقات تدريباً كافياً على كيفية العمل في بيئة قاعدة البيانات بما في ذلك مفاهيم قاعدة البيانات ونمطيات قاعدة البيانات وكيفية كتابة استدعاءات لنظام إدارة قاعدة البيانات وربما كيفية عمل تصميم قاعدة البيانات وكيفية استخدام قاموس البيانات وبصفة عامة الخدمات التي يمكن أن يتوقعوها من إدارة البيانات. وبالطبع يجب أن تقدم وظيفة إدارة البيانات data management تعليماً للعاملين في قسم إدارة البيانات أنفسهم.

الخلافات المتوقعة واستخدام السلطة: بتقديم هذا العنوان يجب أن نستغرق ولو لحظة في فحص السؤال الخاص بملكية البيانات، من الذي يملك جزءاً من البيانات أو ملفاً؟ ولكي تكون الإجابة تقنية حيث أن البيانات هي مورد مرتفع القيمة للشركة فإن البيانات تنتمي إلى ملاك الشركة أو إلى أصحاب أسهم الشركة. أما بالنسبة للاصطلاحات العملية فعادة ما يتحكم المستفيد أو المستفيذ الأولي في البيانات. وفي يوم ما يمكن أن نصل إلى مستوى متطور من المشاركة في البيانات بحيث لا يصبح من الواضح من الذي يمتلك جزءاً معيناً من البيانات، وهنا تقع المسئولية على إدارة البيانات. ولكن الآن عادة ما تعرف البيانات بمستفيد معين وتعمل مجموعة إدارة البيانات كحارس للبيانات بمعنى توفير الأمن والنسخ الاحتياطية وعمل التوجيه ومثل هذه الخدمات الأخرى.

فإذا تطلب تطبيق جديد استخدام بيانات موجودة فعلاً يكون من عمل إدارة البيانات data management العمل كوسيط وتوصيل مالك البيانات بالطلبات على البيانات المشتركة. ويمكن للمالك أن يوافق على هذه المشاركة وعند ذلك يقوم العاملون في إدارة management البيانات بإخطار العاملين في إدارة administration البيانات ليسمحوا للتطبيق الجديد بالاتصال بالبيانات. فإذا كان هناك خلاف عن مثل المشاركة في البيانات هذه فيعمل العاملون في إدارة management البيانات كوسيط بين الأطراف المتنازعة. ويمكن أن يجد العاملون في إدارة management البيانات أنفسهم يعملون كوسيط بين مستفيدين لقاعدة البيانات يستخدمان نفس وحدة التشغيل المركزية مع التنافس لأداء أفضل.

التوثيق والنشر: باستخدام قاموس البيانات على أنه الوسيلة الأولية التي

صمم لها فإن وظيفة إدارة management البيانات تكون مسئولة عن تـوثيق طبيعة البيئة الموجهة للبيانات. ويشمل هذا البيانات الموجودة والملفات والبرامج والتقارير وما إلى ذلك الموجودة هناك ومن يسمح له بالاتصال بهذه العنـاصر وما إلى ذلـك. بعض المعلومات يمكن أن تكون مدخلات بواسطة العـاملين في تطويـر التطبيقـات والعاملين في إدارة البيانات إلا أن مجموعة إدارة البيانات يجب أن تكون قـادرة على تقديم صورة شاملة لذلك.

وكمـوضـوع مـرتبط بـذلـك يجب عـلى العــاملين في إدارة management البيانات أن يقوموا بوظيفة النشر أي أخبار المستفيدين بـالبيانـات الموجـودة فعلاً في قاعدة البيانات. فمعرفة الموجود يمكن أن يشجـع الناس عـلى جعل جـزء أكبر من عملهم آلياً وعلى تكامل عملهم بصورة مباشرة أكثر مع عمليـات الأعمال المـرتبطة به والتي تكون قد حدث لها آلية بالفعل.

إدارة بيئة البيانات: مرة أخرى باستخـدام قامـوس البيانـات يجب أن تكون وظيفـة إدارة management البيانـات مستعدة لـلإجابـة على أي سؤال عن كيفيـة عمل البيئة الموجهة للبيانات. وبمقدرة العلاقات المتداخلة للقاموس يجب أن تكون إدارة management البيانات مستعدة دائماً لـذكر من المسمـوح له بـالاتصال بـأي من الملفات وأي الملفات تستخدم في إنتاج أي من التقارير وأي الـبرامج يستخـدم أي الملفات (ويعرف ذلك بأنه تقديم تأثير التغيير لأن التغيير في ملف يمكن أن يؤثر على برنامج يستخدمه) ومعنوية كل خليط له معنى من كينونات تشغيل البيانات.

مسؤوليات إدارة administration قاعدة البيانات:

توجيه الاستخدام والأمن: العاملون في إدارة administration مسؤولون عن توجيه بيئة قاعدة البيانات على أساس ثابت ونشط. ولعمـل ذلك فـإنهم يظهـرون على أنهم يعملون كتقاطع بين مشغلي الكمبيوتر ومبرمجي النظام وفي الواقع فعادة ما يشار إليهم بأنهم «مبرمجو نظم قواعد البيانات».

أحد الأنشطة الخاصة بهم هو تتبع مسـار من يتصل بـالبيانـات الموجـودة في قـاعدة البيـانات وفي أي وقت يحـدث الاتصال. وعـادة ما تـوجد تسهيـلات نظم برامج تمكن هؤلاء العاملين من أداء هذه الوظيفة ويتم تـوجيه المستفيـدين من بيئة

قاعدة البيانات في الواقع من العديد من وجهات النظر. أحـد هذه الـوجهات هـو موضوع الأمن: التأكد أن الشخص المسموح له بالاتصال بالبيانات فقط هو الذي يتصل بها. ويشمـل ذلك وظيفـة إعطاء تعليـمات للنظام ليسمـح لمستفيدين جـدد بالاتصال بقاعدة البيانات طبقـاً لأوامر العـاملين في إدارة management البيانـات بالاتصال مع ملاك البيانات. وهناك وجه آخر وهو الحاجة إلى حفظ سجلات عن حجم الاستخدام لكل المستفيـدين المختلفين لقـاعدة البيـانات. ويمكن أن يكـون لذلك تـأثير عـلى موازنـة التحميل في المستقبـل وعلى أمثليـة التشغيل كـما يمكن أن يستخدم كذلك في توزيـع تكاليف النـظام على المستفيـدين المختلفين والتـطبيقات المختلفة. وهناك اهتمام مرتبط بذلك وهو عمل مراجعة حسابات لقاعدة البيانات. فبالرغم من افتراض أن المستفيدين المسموح لهم فقط بالاتصـال بقاعـدة البيانـات هم الـذين يفعلون ذلك إلا أنـه هنـاك أسبـاب تشمل المحـاسبة وتصحيـح الخطأ تتطلب سجل لمن يقوم بالاتصـال ومن يقوم بـالتعديـل بأي عنـاصر للبيانـات. فإذا نفذت هذه الوظيفة فإن الوسيلة التي تسمح بتحقيقها هي سجل الأداء log والذي يشبه سجل الأداء المستخدم في عمل النسخ الاحتياطية والاستعادة. وطبقاً لطبيعـة عمل مراجعـة الحسابـات فقد يسجـل سجل الأداء هنـا كل الاتصـالات البسيطة بالبيانات وكذلك كل التعديلات التي جدت على البيانات.

توجيه الأداء وموازنة الأعمال: هناك وظيفة توجيه أخرى ينفذها العاملون في إدارة administration قـاعدة البيـانات وهي تـوجيه الأداء. بـاستخـدام بـرامج التسهيـلات يستطيـع مبرمجـو نظم قـواعد البيـانات تـوصيل أداء التنفيـذ المختلف لبرامج قاعدة البيانات. هذا النشاط له عدد من التـأثيرات. لأحـد الأشياء يكـون مهماً معرفة مدى سرعة تنفيذ التـطبيقات المختلفـة كوظيفـة واضحة لإدارة تشغيـل البيـانات. وهـذه المعلومات تكـون وثيقة الصلة بـالموضـوع كـذلـك لخطط نـظم المكونات ونظم البرامج المستقبلية.

وطبقاً لخواص نظام إدارة قاعدة البيانات كذلك وللنظام الـذي يعمل عليـه يمكن أن تستخدم هذه المعلومات في إعادة توزيع حمل تطبيق قاعدة البيـانات عـلى وحدات تشغيل مركزية مختلفة أو على مناطق ذاكرات مختلفة داخل النظام. وأخيراً يمكن استخدام معلومات الأداء في إخراج التطبيقات غير الكفـوء والتي يمكن إعادة تصميمها.

وهناك ملاحظة إضافية وهي أن مبرمج نظم قاعدة البيانات يجب أن يتداخل مع العاملين المعتادين في منظمة تشغيل البيانات من مبرمجي النظم. ومبرمجو النظم عادة ما يظلوا لديهم مسؤوليات معينة مرتبطة بنظم إدارة قاعدة البيانات مثل إعداد صكوك جديدة لنظام إدارة قاعدة البيانات كما يأتي. كما يمكن أن يكون لمبرمجي النظم المعتادين مسؤوليات أداء ومواجهة مشاكل والتي قد تتداخل مع مبرمجي نظم قاعدة البيانات ونتيجة ذلك هو تسهيل الأمور بكثرة إذا عملت المجموعتان بصورة متعاونة مع بعضهما.

مواجهة مشاكل نظام إدارة قاعدة البيانات: من الحتمي وجود وقت يفشل فيه تطبيق لنظام إدارة قاعدة بيانات أثناء التنفيذ. ويمكن أن يتراوح السبب من خطأ بسيط في شفرة التطبيق إلى فشل في نظم المكونات أو في النظام. والسؤال هو الذي يستدعي المستفيد عندما يحدث ذلك؟ في البيئة المراقبة جيداً يجب أن يعمل العاملون في إدارة البيانات data administration كسطح بيني لمواجهة المشاكل. (لاحظ أنه بينما يعتقد البعض أن هذا العنوان خاص بإدارة management البيانات من إدارة administration البيانات إلا أننا نضعه هنا في جانب إدارة قاعدة البيانات). وعملياً طبقاً لدرجة تعقيد وطبيعة المشكلة يمكن أن يشمل الموضوع أناساً من كل من القسمين.

ومفتاح عملية مواجهة المشاكل هو عمل تقويم الخطأ الذي حدث وتنسيق العاملين اللازمين لفحصه وتصحيحه. وقد يشمل ذلك مبرمجي نظم أو مشغلي معدات أو مبرمجي تطبيقات مع العاملين في إدارة administration البيانات أنفسهم.

عمليات قاموس البيانات: يكون العاملون في إدارة قاعدة البيانات مسؤولون عن العمليات على عكس أوجه التخطيط الخاصة بقاموس البيانات. ويشمل ذلك تطوير طرق لتكون أسطح بينية مع قاموس البيانات بالنسبة لأساليب المدخلات والمخرجات. ويعني ذلك أيضاً توفير اتصال بقاموس البيانات للآخرين مثل محللي النظم وإنتاج تقارير قاموسية دورية طبقاً لطلبات الإدارة وتقديم إجابات للأسئلة الخاصة عن البيانات وبيئة تشغيل البيانات التي تطلبها الإدارة.

السطح البيني مع مشغلي الكمبيوتر: إن وظيفة عمليات الكمبيوتر المستقلة

ما تزال تمثل أدلة كثيرة في بيئة إدارة البيانات وكما هو معتاد يبدأ مشغلوا الكمبيـوتر العمل أو يعيدوا عمل النظام طبقاً للحاجة ويقومـوا بتركيب الشرائط والأقـراص. ويؤدو عمليـات النسخ الاحتيـاطية والاستعـادة وما إلى ذلـك. وأحـد أعـمال إدارة البيانات هـو العمل مـع المشغلين وإعطائهم تعليـمات وتدريبهم عـلى كيفية أدائهم لأعمالهم في الجو الناتج من نظام إدارة قاعدة البيانـات وطبقاً لاحتيـاجات مجمـوعة التـطبيقات. مثـال ذلك يمكن أن تـتراوح هذه الأنشـطة من كيفية بـدء الأعمال في نظام إدارة قاعدة البيانات إلى كيفية تنفيذ استعادة للنظام بعد فشله.

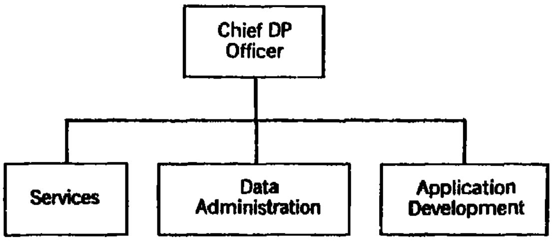
الوضع التنظيمي لإدارة البيانات:

Data Administration Organizational Placement

أين يتناسب وضع إدارة البيانات التنظيمي في تشغيل البيـانات أو في تنـظيم المؤسسة؟ يمكننا أن نبدأ الإجابة على هذا السؤال بأفكـار فلسفية ومستقبليـة بعض الشيء: هل البيانات وتشغيل البيانات عبارة عن عملين مستقلي أو أنهما جزئين من شيء واحد؟ تقليدياً كانت البيانات جزء من تشغيل البيانات. أمـا إذا فكرنـا الآن في البيـانات كمـورد هام للمؤسسـة ألا تكون مـوضوعـاً مستقلاً ويكـون تشغيلهـا موضوعاً منفصلاً؟

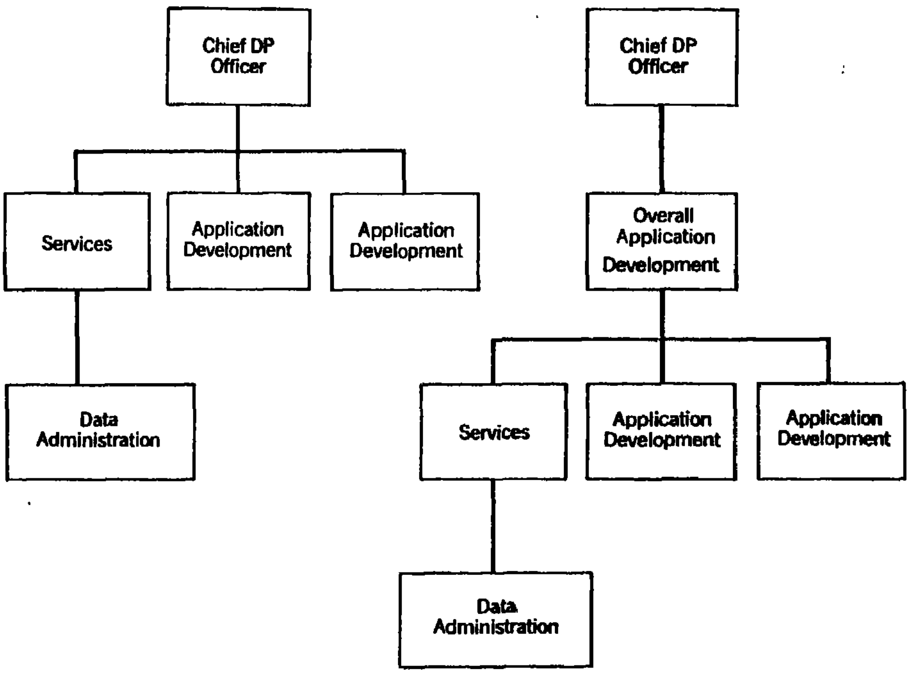
وقبل أن نستطرد دعنا ننظر إلى واقع البيانات وإدارة البيانـات في المؤسسات الموجودة حالياً. في معظم هذه المؤسسات تكون مجمـوعة إدارة البيـانات جـزءاً من قسم تشغيـل البيانـات. وهناك ثـلاثـة مسـارات رئيسيـة لإدارة البيـانـات لعمـل التقارير. أحدهما أن تقدم إدارة البيانـات تقاريـرها إلى الضـابط الرئيسي لتشغيـل البيانات مباشرة (شكل 13.3). والآخر أن تقدم إدارة البيانات تقاريرهـا إلى مديـر الخدمات أو الـدعم التقني والذي يقـوم بدوره بتقـديم التقاريـر إلى رئيس تطويـر التطبيقات أو إلى الضابط الرئيسي لتشغيل البيانات (شكل 13.4). والثالث هـو أن تقدم إدارة البيانات تقاريرها إلى الرئيس الشامل لتطوير التطبيقات مباشرة (شكـل 13.5). والمسـار من إدارة البيانـات إلى الخـدمـات إلى الضـابط الـرئيسي لتشغيـل البيانات هـو الأكثر شيـوعاً من هـذه المسارات الثـلاثة. وإحـدى الحقائق هي أنـه حيث أن وظيفة إدارة البيانـات تخدم مـورداً هامـاً ومشتركـاً فيجب ألا تكون تحت تحكم أي مجمـوعة من مجمـوعات المؤسسـة تتنافس بـدورها عـلى هـذا المـورد مـع

المجمـوعات الأخـرى للمؤسسة. لهـذا فإن وضـع إدارة البيانـات يجب ألا يكـون منحازاً في توزيع الموارد وفض النزاعات التي تحـدث بين المتنـافسين عـلى البيانـات والأداء تحت نظم إدارة قواعد البيانات.



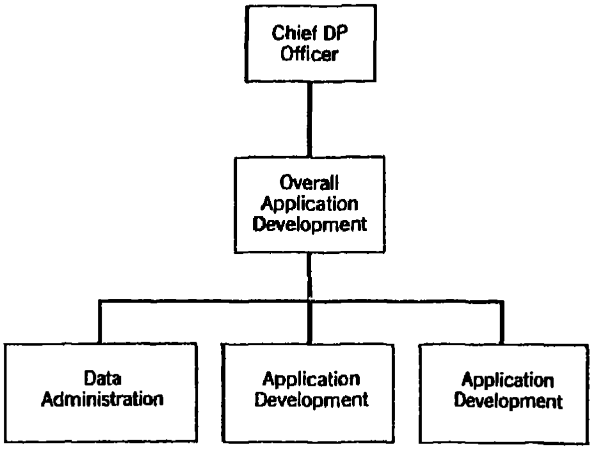
شكل 13.3

إدارة البيانات تحت رئاسة الضابط الرئيسي لتشغيل البيانات



شكل 13.4

إدارة البيانات تحت رئاسة مدير الخدمات



شكل 13.5

إدارة البيانات تحت رئاسة الرئيس الشامل لتطوير التطبيقات

إلا أن هنـاك اختلافـات معنويـة لهذا المـوضوع بصفـة خاصـة في الشركات الكبيرة التي بها مجموعات نشيطة لإدارة البيانات. سـوف تصبح هذه الاختـلافات أكثر اتساعاً مع مرور الوقت ومع زيادة أنشـطة إدارة البيانـات وتشغيل البيـانات. والاختلاف الأكثر شيوعاً هو لشركة لديها مجموعات عديدة لإدارة البيانات. وعـادة ما يحدث ذلك عندمـا تكون الشركـة الكبيرة بهـا تنظيـمات تشغيل بيـانات مستقلة تـدعم أنواعاً مختلفـة أو عمليات مـوجودة في مـواقـع جغـرافيـة مختلفـة أو كـل من الأمرين وكل من تنظيمات تشغيل البيانات المختلفة هذه تقوم بإعداد مجمـوعة إدارة البيانات الخـاصة بهـا بنفسها. ففي هـذه الحالـة يمكن أو لا يمكن أن يوجـد مستوى أعلى لتقديم تقارير وظيفة إدارة البيانات في هذه المستـويات العليـا لتنظيم تشغيـل البيانات أو كذلك خارج محتوى تنظيم تشغيل البيانات. مجموعة تشغيـل البيانـات مرتفعة المستوى هذه يمكن أن تتغير من مجموعة إعداد نمطيات بيانات للمؤسسة مع فقدان التنسيق الى مجموعة قوية يمكنها إمـلاء اجراءات محـددة وتطبيقـات مشتركـة كذلك للمواقع المختلفة.

التأثير التنظيمي لإدارة البيانات :

Data Administration Organizational Impact

لا يمكن أن ينفـذ منهج إدارة البيـانـات في يـوم وليلة. فهـو يشمـل داخلياً تغييرات في مسؤوليات العمل لمطوري التطبيقات ولآخرين وتغييرات في اجـراءات تشغيـل البيانـات بصفة عـامة. وأقسـام إدارة البيانـات الجديـدة تجد مقـاومـة من المـديرين الـذين لا يفهمون هـذا المعنى ومن مطوري التـطبيقات الـذين يرتـابـون بطريقة طبيعية للتغييرات التي تطرأ على مسؤوليات عملهم وللطرق الجديدة لإنتاج البرامج ومن المستفيـدين الذين يشملهم مفهـوم مشاركـة الآخرين لهم في بيـاناتهم الخاصة بهم. وتاريخياً رأينـا أنه هنـاك منحنى تعلم مصاحب لإدارة البيـانات لكـل المعنيين وإن مجموعات إدارة البيانات الناجحة تتطور تدريجياً في تحملهـا مسؤوليات أكثر.

كيف تختلف بيئة تشغيل البيـانات بعـد إدخـال وتـطويـر إدارة البيـانـات؟ بتركيزها على التخطيط فإن إدارة البيـانات تنتـج حرصـاً على العـلاقات المتـداخلة لبيانات المؤسسة. إنها تتسبب في أن العاملين الآخرين في تشغيل البيانات يقـومون بالدراسة لعمليات الأعمال بدقـة أكبر في محـاولة لجعلهـا تؤدي آلياً. ويؤخـذ تحليل النظم بجدية أكثر عن ذي قبـل ومن المحتمل أن تـوجه الإدارة وقتـاً أكثر من وقت تطوير التطبيق لتحليل النظم.

ومـع امتصاص إدارة البيـانات للمـزيد من العمـل المكثف للبيـانـات يمكن لمبرمجي التطبيقات أن يركزوا على أوجه المنطق لأنشطة برامجهم. هذا بالإضـافة إلى تحسين تصميم قاعدة البيانات.وبرامـج مواصفات البيانـات الأكثر دقـة التي ينتجها المتخصصون في إدارة البيانات تقود إلى عملية تطويـر للتطبيقـات أقل أخـطاء وإلى برنامـج نهائي أكثر تمـاسكاً. كـما أن جودة وتمـاسك البيـانات في قـواعد البيـانات تتحسن مع قلة التكرار وتحسن سلامة البيانات.

وبصفة عامة فإن التركيز المتزامن على البيانات كمورد يضع محتـويات عمليـة تشغيل البيانات في الشركة في ضوء جديد في عيون إدارة المؤسسة. فتتحقق الإدارة أن البيانات هي رأس مال يمكن استخدامه في تحسين مـوضع المؤسسـة التنافسي في

أسواقها. ويوضع تركيز أكثر على تشغيـل البيانـات وتركيـز أكبر عـلى مقدرة قسم تشغيل البيانات في أن يجعل المزيد من عمليات أعمال المؤسسة آلية.

إدارة البيانات وتقنيات الحسابات الجديدة:

Data Administration and New Computing Technologies

حديثاً وجدت تطورات جديدة في تقنية تشغيل البيانات وكلها مرتبطة بعض الشيء ببعضهـا ومفهـوم إدارة البيـانـات. أحـد هـذه التـطورات هـو الكمبيـوتـر الشخصي. وتطور آخر هـو لغات الجيـل الرابـع. وهذه تقـدم تطويـر للتطبيقـات مرتفع المستـوى عن اللغات النمـطية مثـل الكوبـل والبسكال والتي تعتـبر مناسبـة ليستخدمها المبرمجون وغير المبرمجون. وهناك تطور ثالث وهو مفهوم مركز المعلومات والـذي يوفـر للمبرمجـين وغير المـبرمجين عـلى حد سـواء اتصالاً بعمليـة استخـدام الكمبيـوتر من خـلال لغات الجيـل الرابـع ويقدم المسـاعدة عن طـريق استخـدام الكمبيوتر الشخصي ووسائل أخرى تتجنب عملية تطوير تطبيقات تشغيل البيانات المعتادة.

والعلاقة بين مركز المعلومات بوسائله مـع إدارة البيانـات واضحة. فلغـات الجيل الرابع وأجهزة الكمبيـوتر الشخصيـة ووسائـل الكمبيوتـر الأخرى الصـديقة للمستفيد تتطلب بيـانات منـاسبة لأداء العمـل. فالمستفيـد الذي يستخـدم إحدى هذه الوسائل ويمكنه الاتصال بقسم إدارة البيـانات والحصـول على البيـانات والتي كان في حاجة إلى جمعها بنفسه يكون قـد حصل عـلى مساعـدة كبيرة في أداء عمله ولا يكون هذا أمر هشاً عند هذه النقطة. فالبيانات مـورد يجب أن يدار بمثـل هذه الطريقة ويجب أن يكون متاحاً للمسموح لهم في الشركة بالحصول عليها. ومـركز المعلومات ولغات الجيل الرابع وأجهزة الكمبيوتر الشخصي تـوفر الـوسيلة للعديـد من الأفـراد المختلفين في المؤسسـة لاستخدام البيـانات التي تـديرهـا وظيفـة إدارة البيانات لتحقيق أهداف الشركة.

إدارة مورد المعلومات: Information Resource Management

إن التخيل بالنسبة للمستقبل الخاص بإدارة البيانات عبارة عن تمرين يتحول بسرعة إلى التخيل الى مستقبل تشغيل البيانات. مرة أخرى علينـا أن نسأل «أيهـما

يـأتي أولاً البيانـات أم التشغيل؟» تـاريخياً كنـا مهتمين بـالتشغيـل لـدرجـة إغفـال البيانات. وهذا الترتيب بدأ يتغير ببطء مـع بدء استخـدام وانتشار استخـدام نظم إدارة قواعد البيانات.

ويمكننا أن نجد بـالفعل ضبـاط رئيسيين لتشغيـل البيانـات لهم ألقاب مثـل نائب الرئيس لموارد المعلومات. وقد يرى المستقبل تغلغلا لذلك في المستوى الأعلى وخلال محتويات تنظيم تشغيل البيانات إلى النقطة التي تتميز فيها البيانات على أنها اهتمام حرج ورؤية تشغيلها على أنه ضرورة ميكـانيكية أكـثر من أنه غـاية في حـد ذاته.

وكجزء من هذا التحويل سوف نرى المزيد من البيانات والتطبيقات التي يتم إدخالها في بيئة قاعدة البيانات. هذه البيانات والمزيد من البيانات غـير الموجـودة في قاعدة البيانات سوف تقع تحت تحكم أو تحت تأثير تنـظيم إدارة البيانـات. وسوف يحفظ المزيد من أنواع البيانات مثل البيانات السياسية في قواعد بيانات المؤسسات. كـما سيوجـد نمو مـتزايد في اتجـاه البيانـات المشتركـة وقلة التكرار وقـواعد بيـانات للشركات مخططه جيداً ومتكاملة بصورة طيبة. وسوف نصل إلى بيئـة نستطيـع أن نسـميها بالفعل بإدارة مورد المعلومات Information Resource Management.

مراجع: References

Durell, W.R., Data Administration, McGraw-Hall, New York, 1985.

Gillenson, M.L., «The State of Practice of Data Administration - 1981», ACM Communications, vol. 25, no. 10, October, 1982, pp. 699 - 706.

Gillenson, M.L., «Trends in Data Administration: 1981 - 1983», IBM Systems Research Institute Technical Report -- TR73 - 027, New York, January, 1984.

Gillenson, M.L., and Goldberg, R., Startegic Planning, Systems Analysis, and Database Design, Wiley, New York, 1984.

GUIDE International Corp., «Establishing the Data Administration Function», Chicago, 1977.

GUIDE International Corp., «Data Administration Methodology», Chicago, 1978.

Lyon, J.K., The Database Administrator, Wiley, New York, 1976.

Martin, J., Managing the Data-Base Environment, Prentice-Hall, Englewood Cliffs, NJ, 1983.

McCririck, I.B., and Goldstein, R.C., «What Do Data Administrators Really Do?», Datamation, vol. 26, no. 8, pp. 131 - 134, August, 1980.

Ross, R.G., Data Dictionaries and Data Administration, AMACOM, New York, 1981.

Weldon, J.L., Data Base Administration, Plenum Press, New York, 1981.

أسئلة وتمارين: Questions and Exercises

١ ـ اختر أحد موارد المؤسسة التقليديـة (مثل النقود أو المباني والمعدات أو الأفراد وخلافه) واعمل تخطيطاً للهيكل التنظيمي للمؤسسة الذي يدير هذا المورد.

٢ ـ اختر أحد موارد المؤسسة التقليديـة (مثل النقود أو المباني والمعدات أو الأفراد وخلافه) وصف تأثيراته على المؤسســة إذا لم يكن هناك مجمـوعة مسؤولـة عن إدارته.

٣ ـ صف أهمية المشاركة في البيانات من وجهة نظر الإدارة.

٤ ـ كيف يمكن أن تكون البيانات مشتركة بين:

أ ـ أقسام الأفراد وإدارة المعدات.

ب ـ أقسام المحاسبة والتشغيل.

جـ ـ أقسام التشغيل والأبحاث.

د ـ أقسام المحاسبة وتشغيل البيانات.

٥ ـ إقنع إحدى الشركات بأن تنشيء قسم لإدارة البيانات.

٦ ـ صف كيف يمكن لوظيفة إدارة البيانات أن تغير من أعمال الآخـرين في قسم تشغيل البيانات.

٧ ـ إعمل تخطيطاً لمسئوليـات جزء إدارة البيـانات data management من قسم إدارة البيانات data administration ـ أي من هـذه المسؤوليات تعتقـد أنه أهمها؟ وما سبب ذلك؟

٨ ـ كيف يجب أن تنسق وظيفة تخطيط البيـانات مـع وظائف التخـطيط الأخرى بالمؤسسة؟

٩ ـ كيف يكـون ردك على مـبرمج أو محلل نظـم يعتبر دور إدارة management البيانات أنه تطفلاً في مهنته؟

١٠ ـاعتـبر التمييز بـين ملكية البيـانات والحـراسـة للبيانات. ما مـدى شعـورك بـالارتياح لمسؤوليـة العاملين في إدارة البيـانات عن فض النـزاعات الخـاصة بالمشاركة في البيانات؟

١١ ـإعمل تخطيطاً لمسؤوليات جزء إدارة قاعدة البيانات من قسم إدارة البيانات، أي هذه المسؤوليات تعتقد أنه أكثر أهمية؟ وما سبب ذلك.

١٢ ـكيف يكون ردك على مبرمج أو محلل نظم يعتبر دور إدارة قاعدة البيانات أنه تطفلاً في مهنته.

١٣ ـصف تجزئة العمل الخاصة بقاموس البيانات بين أفـراد إدارة management البيانات وأفراد إدارة قاعدة البيانات.

١٤ ـأسرد مجموعة من الحجج لوضع وظيفة إدارة البيانات:

أ ـ مرتفعة في تنظيم تشغيل البيانات.
ب ـ خارج تنظيم تشغيل البيانات.
جـ ـ جزئياً داخل تنظيم تشغيل البيانات وجزئياً خارجه.

١٥ ـصف تأثير إدخال مجموعة إدارة البيانات في تنظيم تشغيل البيانات.

١٦ ـهل تعتقد أن إدارة البيانات يجب أن تكون مسؤولة عن البيـانات المحفـوظة في أجهزة الكمبيوتر الشخصي أو أجهزة الميكروكمبيوتر الخاصة بالشركة؟ لماذا ولم لا؟

١٧ ـهـل تعتقد أن بقيـة تنظيم تشغيـل البيانـات يقدم تقـاريـره إلى تنـظيم إدارة البيانات؟ لماذا ولم لا؟

الفصل الرابع عشر
مواضيع متطورة
Evolving Topics

مقدمة : Introduction

نقدم في هذا الفصل ثلاثة مواضيع في قاعدة البيانات تعتبر جديدة نسبياً أو تعتبر تجريبية نسبياً أو كلاهما. فآلة قاعدة البيانات database machine هي حل لنظم المكونات لتحسين الأداء في بيئة قاعدة البيانات. وتمثل قاعدة البيانات المنتشرة distributed database مفهوم قاعدة البيانات المقسمة إلى أجزاء ومخزنة في نقاط مختلفة من نقاط شبكة كمبيوتر. وقاعدة البيانات على أجهزة كمبيوتر شخصية database on personal computer تشير إلى المشاكل الخاصة التي تظهر عند محاولة تنفيذ مفاهيم قاعدة البيانات على آلات كمبيوتر صغيرة.

آلات قواعد البيانات : Database Machines

مفهوم آلة قاعدة البيانات:

لقد ركزت مناقشاتنا في هذا الكتاب حتى الآن على نظم البرامج والبيانات على عكس نظم المكونات. وفيما عدى الوصف المتقدم لوحدات الأقراص فقد تكلمنا عن نظم إدارة قواعد البيانات وعن قواميس البيانات ومفاهيم البيانات وتكوينات البيانات وغير ذلك إلا أنه هناك فرع من فروع قاعدة البيانات جديد ويهتم بنظم المكونات المتخصصة. وقد أصبح هذا الفرع مسمى بآلات قواعد البيانات database machines.

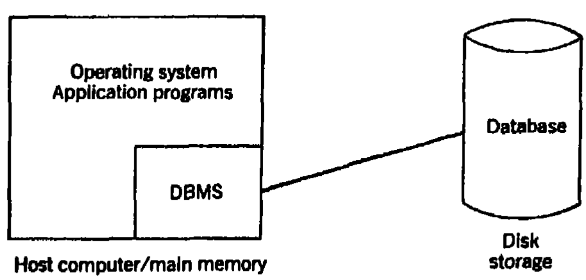
وكما سوف نرى في مناقشة المناهج المختلفة فإن العمل في آلات قواعد البيانات تقدم في العديد من الاتجاهات المختلفة. وكانت الدوافع له تشمل ما يلي:

* لتحرير المشغل المضيف من مسؤولية الاتصال بالبيانات.
* لتحسين الأداء في البيئة الموجهة لقاعدة البيانات.
* للسماح بالاتصال بالبيانات بواسطة مشغلات مضيفة متعددة.
* لتحسين أمن وسلامة البيانات.

دعنا نلقي نظرة على المفاهيم ونناقش بعض مميزاتها وعيوبها ثم نستخلص الخبرة الحالية.

مناهج آلة قاعدة البيانات:

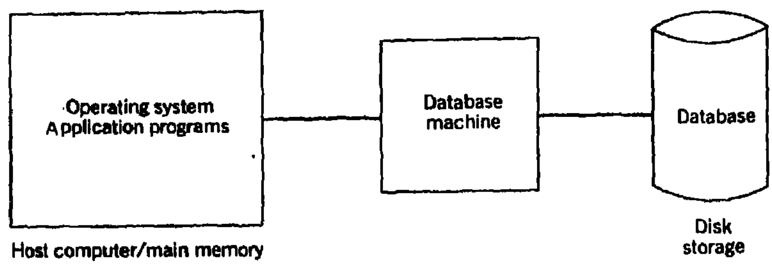
مشغل نقطة النهاية التقليدي: يوضح شكل 14.1 بيئة تقليدية لقاعدة بيانات يكون فيها الكمبيوتر المضيف مسؤولاً عن تشغيل نظام إدارة قاعدة البيانات. وتشمل مسئولية الكمبيوتر المضيف حفظ كل نظام إدارة قاعدة البيانات أو جزء منه في الذاكرة الرئيسية وتشغيل طلبات البيانات المخزنة في قاعدة البيانات واتصالات هذه الطلبات بوحدات الأقراص عبر مراقباتها. وبعض الناس يعتقدون أنه بنقل هذه الوظائف إلى جهاز منفصل فيمكن للكمبيوتر المضيف أن يقضي وقتاً أكثر من وقته في أداء أنشطة تشغيل أخرى ويمكن إشباع طلبات البيانات بكفاءة أعلى.



شكل 14.1
بيئة تقليدية لقاعدة البيانات

ويبين شكل 14.2 بيئة قاعدة بيانات تستخدم مشغل نقطة نهاية تقليدي كآلة قاعدة بيانات. وقد استخدم جهاز كمبيوتر أصغر ومستقل بذاته بين

الكمبيوتر المضيف ووسط تخزين القرص. وآلة قاعدة البيانات هي ببساطة كمبيوتر تقليدي مشكل لأداء وظيفة واحدة فقط: وهي تشغيل نظام إدارة قاعدة البيانات. تـأخذ الآلـة طلبات البيـانات من الكمبيـوتر المضيف في صـورة استـدعـاءات من البرنامج أو استفسارات خـاصة من نهايـة طرفيـة وتترجم هـذه الطلبـات إلى أوامر اتصال وتسترجع البيانات من وسط تخزين القرص ثم تعيد البيانات إلى الكمبيـوتر المضيف.



شكل 14.2

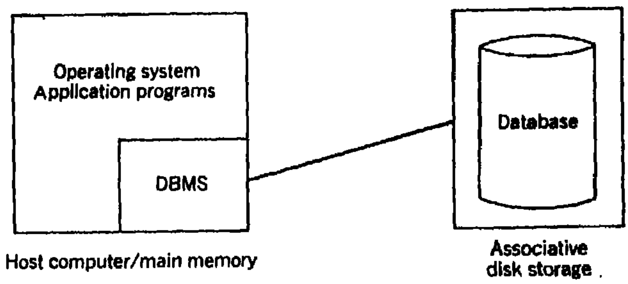
بيئة قاعدة بيانات بمعضل خلفي تقليدي

الـذاكرة المصـاحبة: لقـد اقترح حـل نظم مكـونات لعـدد كبير من عمليـات المقـارنة لتكـون في وجه داخـلي لنظام عـلاقي لإدارة قاعـدة البيانـات. والفكرة هي تصميم نظم مكونات لتخزين بيانات ذات أغـراض عامـة تكون قـادرة على تنفيـذ عملية معطاة على أجزاء مختلفة من ملف في قاعدة البيانات في نفس الوقت. مثـال ذلك قيمة حقـل معطى في كـل سجل من ملف يمكن أن تقـارن مع قيمـة معينة. هذا النوع من العمليات يكون شائعاً بصفة خاصـة في بيئة النـظم العلاقيـة لإدارة قاعدة البيانات. وعلى هذا لا يكون من المرغوب فيه من وجهـة نظر الأداء أن تكـون هناك مقدرة على اختبار الحقل في العديد من هذه السجـلات مع بعضهـا ومعالجـة النتائج طبقاً لذلك بدلاً من تشغيلها واحداً واحداً منذ بداية الملف وحتى نهايته؟

وحدات الذاكرة المصاحبـة والمعروفـة كذلـك بأنها مشغـلات متوازيـة ذات عنـاوين لمحتويـاتهـا content addressable parallel processors يمكن أن تشيـد بطرق عديدة. وكمثال تخيل مشغل أقـراص له رأس قـراءة وكتابـة ثابت عـلى كل

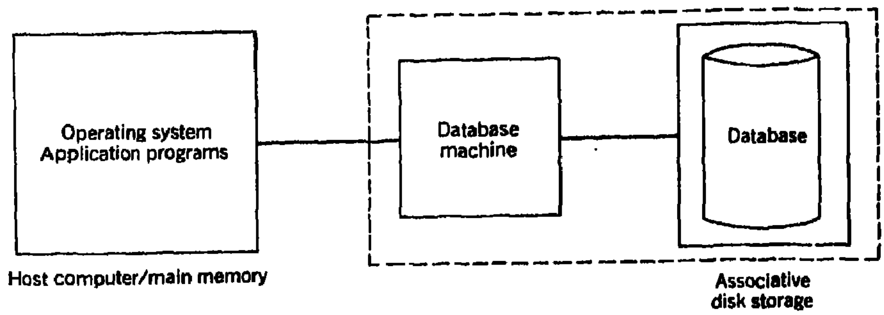
مسار لكل اسطوانة. وتخيل كذلك مشغل دقيق موجود في كل رأس من رؤوس القراءة والكتابة هذه. عندما يحدث طلباً للاتصال ببيانات ويشمل نوع عملية المقارنة الذي سبق وصفه الآن فيمكن لكل من هذه المشغلات الدقيقة أن تأخذ اتجاهات في نفس الوقت لأداءة العملية. افتراضاً إذا كانت سرعة تشغيل المشغلات الدقيقة هذه سريعة بدرجة كافية بالنسبة إلى سرعة دوران القرص فيمكن عند ذلك فحص كل سجل في الملف وتخزن في مجموعة الأقراص هذه في دورة واحدة كاملة. وهناك صيغ أخرى يمكن أن يكون بها عدد أقل من المشغلات الدقيقة تغطي أجزاء أكبر من الملف المخزن. مثال ذلك يمكن أن يكون هناك مشغل دقيق واحد في كل أسطوانة مع فحص كل السجلات الموجودة على الأسطوانة بواسطته خلال العديد من دورات مجموعة الأقراص.

ويبين شكل 14.3 ترتيبين ممكنين لمساهمة الذاكرة المصاحبة داخل بيئة قاعدة البيانات. يقترح شكل 14.3a أنه لا يوجد مشغل قاعدة بيانات منفصل مثل ذلك ولكن تكون الذاكرة الثانوية هي ببساطة الذاكرة المصاحبة. أما شكل 14.3b فيحدد أنه هناك مشغل قاعدة بيانات منفصل والذي يمكن أن يكون أو لا يكون جزءاً من نفس الوحدة الواقعية مثل الذاكرة المصاحبة (وهذا هو سبب الخطوط المتقطعة).



شكل 14.3a

بيئات قواعد بيانات : ذاكرة مصاحبة بدون مشغل منفصل لقاعدة البيانات



شكل 14.3b

بيئات قواعد بيانات: ذاكرة مصاحبة بمشغل منفصل لقاعدة البيانات

مناهج أخرى لآلة قاعدة البيانات: لا يوجد اتفـاق واضح بـين العاملين في هذا المجال عما يكون أو لا يكون آلة قاعدة البيانات. ومن الأمـان القول أن كـل شخص يعتبر المنهجين الذين سبق ذكرهما يقعان داخل مفهوم آلة قاعدة البيانات. والبعض يعتقد أنهما هما المنهجـان الوحيـدان إلا أن البعض الآخر يمكن أن يضيف إليهما من الوحدات أو الترتيبات التالية:

أولاً: هنـاك عقدة الشبكـة network node: والفكرة هنـا هي وجـود آلـة قاعدة بيانات تخدم كعقدة في شبكة أجهزة كمبيوتر منتشرة. وقـاعدة البيـانات التي توجه بواسطة آلة قاعدة البيانات هذه تكون متاحة لأي مشغل مـوجود في الشبكـة ومسموح له باستخدامها.

ثانياً: هنـاك هرميـة التخزين storage hierarchy: يتم تـرتيب سلسلة من وحـدات تخزين البيـانات بحيث أن إحـدى نهايات السلسلة تكـون كميـة صغـيرة نسبياً من تخزين سريع جداً ومكلف جـداً بينما يـوجد عنـد النهايـة الأخرى كميـة كبيرة نسبياً من تخزين بطيء نسبياً لكنه رخيص نسبياً. ومعظم البيـانات المـوجودة في قاعدة البيانات تخزن في النهاية البطيئة الرخيصـة للسلسلة أما البيـانات النشـطة لتطبيق معين فتـدفع إلى (تنسـخ في) نهايـة الـطيف الأسرع وذلـك لـلاسـترجـاع السريع. ويمكن في واقع الأمـر أن يذكـر أحد الأفـراد أن خليط الذاكـرة الرئيسيـة المعتادة وذاكرة التخزين الثانوي يمثلان نوعاً محـدوداً من هرميـة التخزين. وخـطوة أكثر عمقاً في نفس الموضوع هي تسلسل الذاكرة الرئيسية ووحدة الأقراص ووحدة التخزين الوفير.

وأخيراً فإن بعض العاملين في هذا المجال يعتبرون نظام الكمبيوتر مثل نظام IBM (System/38) الذي سبق مناقشته في الفصل التاسع بأنه آلة قاعدة بيانـات. واستخـدام هذا الاصطلاح لا يشير هنـا إلى المعالم التي سبق لنـا مناقشتهـا بل إلى البيئة الموجهة إلى قاعدة البيانات أكثر والتي يصمم لها مثل هذه الأجهزة لتدعمها.

تحليل آلات قواعد البيانات:

مميزاتها: لقد اقترح مدى واسع من المميزات كتبرير لاستخدام آلات قـواعد البيانات. الكثير منها تخيـلي أو يمكن أن يحدث جـدل بالنسبـة له. وبعضهـا يعتمد على التكاليف النسبية للتغير السريع في نظم المكـونات بينـما يعتمد البعض الآخـر على كفاءة تنفيذ الملفات لمنهج معين لآلة قاعدة البيانات.

والسبب الأكثر وضوحاً لتشييد آلة قاعـدة البيانـات هو تحسـين أداء النظام. ولسوء الحظ أن حقيقة إضافة مستوى جديد من الاتصالات تضاف مع آلة قاعـدة البيانات بين قاعـدة البيانـات والكمبيوتـر المضيف سوف تقلل من زمن الاستجـابة لأي استفسار معين من قاعدة البيانات. وأي تحسين في الأداء يأتي من الحقيقة بأن العمليـات التي تشمل أكـثر من استفسار واحـد يمكن أن تنفذ في نفس الـوقت مع عمل الكمبيوتر المضيف وآلة قاعدة البيانات مستقلين. بالطبع هذا يحضر نقطة أن الطريقة الوحيدة التي تجعل آلة قاعدة البيـانات ذات معنى في المقـام الأول هي أنه إذا استخـدم الكمبيـوتـر المضيف في معـظم الأحيـان في دعم نـظم إدارة قــاعـدة البيانات والاستفسارات التي يتم تشغيلها بواسطته. وهناك نقطة أخرى عن تحسين الأداء أن مشغل آلة قاعدة البيانات والمصاحب والمعقد بدرجة كـافية سـوف يسرع من نـوع عمليات البحث التي سبق لنـا وصفها والتي تكـون بعـد كـل ذلـك هي الهدف المقصود.

وحقيقة أن آلة قـاعدة البيـانات تؤدي عمـلاً كان يؤديـه الكمبيوتـر المضيف تحرر الكمبيوتر المضيف بطريقتين. فهي تحرره من معظم أو كل مسئوليات تشغيـل قـاعدة البيـانات. بمـا في ذلك تشغيـل الاستفسارات وحفظ الملفـات وتسمـح لـه بتـوجيه وقت التشغيـل إلى المستفيدين الآخـرين. كما أنها تحـرر مـواقـع الـذاكـرة الرئيسية التي كانت تحتوي على أجزاء نظام إدارة قاعدة البيانات أثناء التنفيذ.

ويمكن لآلة قاعدة البيانات والبيانات التي تتحكم فيها أن يشترك فيها العديد

من أجهزة الكمبيوتر المضيفة. وبـالمثل يمكن أن تكـون آلة قـاعدة البيـانات نقـطة بيانات في شبكة تشغيل منتشرة. وفي الواقع فإن أجهزة الكمبيـوتر المضيف ونـظام براجمها لا تحتاج لأن تكون ذات تشكيل معين إذا كانت آلة قاعدة البيـانات قـادرة على ترجمة استفسارات الصيغ المختلفة الى شكل معين للتشغيل.

ويمكن لنظـم المكونات في آلة قاعدة البيانات أن تكون متخصصة ولا يكـون هناك حاجة لعناصر مثل مشغلات العقدة المتحركة. كما أن نـظم المكونـات يمكن أن تكون من تقنية أكثر تقدماً عن الكمبيوتر المضيف ويسمح ذلك بتحسين في الكفاءة دون استبدال الكمبيوتـر المضيف. وافترضيـاً يمكن تأجيـل الحصول عـلى كمبيوتـر مضيف أكـبر وأحدث بسبب السعـة الموجـودة حديثاً في الكمبيوتـر الحـالي بسبب سحب بعض الأحمال عنه وأسنادها إلى آلة قاعدة البيانات.

وتشمل الحجج الإضـافية أمن البيـانات وسـلامتها واستعـادتها. وحيث أن البيانات التي يتم التحكم فيها بواسطة آلة قـاعدة البيـانات يمكن الاتصـال بها من خلال سطح بيني فردي فقط فيمكن أن يكون التركيز شديد على مجهود الأمن. ولا يوجد خوف من أن يلتف شخص معين حول نظام إدارة قـاعدة البيـانات للحصـول على البيانات كما هـو الحال في الـترتيبات المعتـادة. وبالـطبع مـا زال هناك ضرورة لوجود طريقة للحماية ضد ظهور شخص غير مسموح له بأن يدخل إلى آلة قاعـدة البيانات من خلال سطح بيني نمطي. وعلى نفس الخطوط يمكن التأكد من سلامـة البيانات بصورة أفضل. وكما في حالة الاستعادة فمن الأفضـل عزل الفشـل الذي يحـدث في الكمبيوتـر المضيف عن قاعـدة البيانـات بالـرغم من أن الفشل في آلـة قاعدة البيانات يجب أن يظل متصلاً بالكمبيوتر.

العيوب: الحجة العكسية لميزات الأداء سبق أن ذكـرت بالفعـل كمستوى اتصالات إضافي ضروري للاستفسار الفردي. بالإضافة إلى ذلك فهناك مـواضيع أخرى خاصة بحقيقة أن مشغلين مختلفين (الكمبيوتر المضيف وآلة قاعدة البيانات) موجودين في هذا الترتيب يمكن أن يعني منهجاً أكثر تعقيـداً للمشغلين وتكلفه أكـثر في التطوير والصيانة وزيادة في التدريب والـدعم وإمكانيـة تقليل الاعتـمادية نـظراً لوجود آلتـين يمكن أن يحدث فشـل بهما. وإذا كـانت الآلتان تم صنعهـما بواسـطة شركتـين مختلفتين فيمكن أن يمثـل التنسيق بينهما مشكلة. وبـالمثل فقـد تكلمنا عن

مميـزات اللغات المتعـددة التي ترسـل استفسارات لها أشكال مختلفـة لآلة قـاعـدة البيانات إلا أن هذا يمكن أن يكون مشكلة معنـوية للعمـل كذلـك. ففي الواقـع يمكن أن يتطلب الاستفسار الفريد من آلة مضيفة واحدة مجهوداً معنوياً في التحويل الى صورة تفهمها آلة قاعدة البيانات.

وهناك بعض الحجج السلبية الأخرى والتي تشمـل: حيث أن نظم الـبرامج يتزايد رخص ثمنها بصفة عامة فالتركيز على أن منهج آلة قـاعدة البيـانات يـوفر في الأمـوال من خلال الاستخـدام الأفضل لنـظم المكونـات يصبح غـير ذو قيمة. في الـواقع قـد يكون من الأبسط والأرخص التـوسع في سعـة الكمبيوتـر الرئيسي عن إدخال آلة قاعدة بيانات. فالمشغلان ـ كمبيوتر مضيف وآلـة قاعـدة بيانـات ـ تقل مرونتهما عن المشغل المتعدد العـام، لأن التخصص في آلة قـاعدة البيـانات يعـوق إمكـانية مـوازنة الحمـل عبر المشغـلات المختلفة. أخيـراً إذا اشتركت عـدة أجهزة كمبيوتر مضيفة في آلة قاعدة بيانات في نفس الوقت فقد يصبـح السريان للبيـانات من وإلى آلة قاعدة البيانات مماثلاً لعتق زجـاجة يبـطيء من سرعة كـل المشغلات. خلاصة ذلك، عند كتابة هذا الكتاب كان هناك العديد من آلات قواعد البيـانات في السوق وغيرهـا في المرحلة المختلفـة من التجارب تمثـل كل الفئـات الرئيسيـة. وهناك من يعتقدون في أنهم يستحقون كما أنه هناك المتشككـون بالنسبـة لتقيم هذه الآلات وهنـاك كذلـك الذين يفكـرون أن هـذه الآلات تعمـل جيـداً في مـواقف تطبيقات معينـة ولا تعمل في مـواقف أخرى وآخـرون يعتقدون أنـه هناك ضرورة للتقدم المفاجيء في سرعة وتكلفة نظم المكونات (وبصفة خـاصة في حـالة المشغـل المصاحب) قبل إمكانية الدخول في داخليـاتها. وبـالقياس بحجم الاستخـدام فإن آلات قواعد البيانات ليست عاملاً معنوياً في سوق قواعد البيانات حالياً إلا أنها قد تصبح كذلك في المستقبل.

قواعد البيانات المنتشرة: Distributed Databases

مفهوم قاعدة البيانات المنتشرة:

يحتوي تصميم نظام الكمبيوتر في الصورة المركزية المعتادة على مشغـل فردي مع وحدات تخزين البيانات المصاحبة له. ويقدم النظام امكانيات تشغيـل بيانـات للمستفيدين الموجـودين في نفس الموقـع أو في مواقـع جغرافيـة متشتته عـبر نهايات

طرفية. وبينما يمكن التحكم في إدارة النظام وبياناته بطريقة مباشرة في هذا الترتيب إلا أنه هناك مشـاكل كبـيرة مصاحبـة لـه كـذلـك. مثـال ذلـك إذا تعـطل المـوقـع المركزي فيصبح كل فـرد ممنوعـاً من استخدام الكمبيـوتر حتى يتم تصليـح العطل ويعمل الكمبيوتر مرة أخرى. كما أن تكاليف الاتصالات من النهايات الطرفية إلى المـوقع المـركزي يمكن أن تكـون مرتفعـة. أحد حلول مثـل هذه المشكلة وبـديـل تصميم لمفهوم التشغيل المركزي كذلك هو التشغيل المنتشر.

في نظام التشغيل المنتشر أو نظام الكمبيوتـر المنتشر توضـع أجهزة الكمبيـوتر في مـواقع جغـرافية مختلفـة ويمكن اتصالها ببعضهـا البعض لتسهيـل العـديـد من أشكال التشغيل. مثال ذلك يمكن نقل نشاط تطبيق فعلي إلى مشغل حر بـدلاً من انتـظاره لمشغل مشغـول حتى يفرغ ممـا هو مشغـول به. إلا أن العـديد من أوجـه استخدام الكمبيوتر المنتشر يشمل طرق تخزين البيانات في الشبكـة بصورة مبـاشرة ويتم الاتصال بها بواسطة برامج التطبيقات. ونتيجة لذلك فإن محتوى حقل جزئي للتشغيل المنتشر ـ ولقاعدة البيانات ـ أصبح مـوجوداً تحت اسم البيـانات المنتشرة distributed data أو قاعدة البيانات المنتشرة distributed database.

وحيث أن قـاعـدة البيـانـات المنتشرة هي إحـدى حـالات المـوضـوع فمن الصعب تحديد تعريف دقيق لها. ولكن بصفـة عامـة هي عبارة عن تـرتيب تكون فيه البيانات الموجـودة في مواقـع مختلفة (أو نقـاط مختلفة) في نـظام أجهزة كمبيـوتر منتشر (أو شبكـة منتشرة) يمكن الاتصـال بهـا بـواســطة الـبرامــج أو المستفيـدين الموجودين عند نقاط أخرى من نقاط الشبكة. وهناك طرق مختلفة عـديدة يمكن أن يتم تشغيل هذه البيانات المنتشرة بها.

في إحـدى نهايات الـطيف يوجـد نظام منتشر يسمـح بتنفيذ الـبرنـامـج عنـد إحدى العقد للاتصال ببيـانات عنـد عقدة أخـرى. وتوجـد مثل هـذه النظم حـالياً كنظم تجارية فعلاً. وعلى سبيل المثال فإن قواعد البيـانات DL/I المـوجودة عنـد عقد من عقـد شبكة منتشرة يمكن الاتصـال بها من عقـد أخـرى خـلال إحـدى السـمات المسـماة الازدواج المتعدد النـظم Multiple Systems Coupling (MSC) في بيئـات IMS/DC أو من خلال إحدى سمات المسماة اتصالات ما بين النظم Intersystem Commuication (ISC) في بيئات CICS. وفي حالة MSC يرتب البرنامج الموجود

في إحدى العقد تنفيذ البرنامج الموجود عنده البيانات المطلوبة والذي يقوم بالاتصال بالبيانات. وترسل بعد ذلك النتائج إلى العقدة التي طلبتها. وفي حالة ISC فإن البرنامج الموجود عند إحدى العقد يطلب بصورة مباشرة البيانات من قاعدة بيانات DL/I الموجودة عند عقدة أخرى في العمل التالي.

وعند النهاية الأخرى للطيف يوجد موقف أكثر تعقيداً جداً توجد فيه الملفات اللازمة للرد على استفسار معين يشمل وصلاً علاقياً في عقد مختلفة من الشبكة. وبطريقة أو بأخرى يتم إحضار البيانات مع بعضها (أو يتجزأ الاستفسار أو كلا منهما) وينفذ الوصل وترسل النتيجة الى العقدة التي أصدرت الاستفسار. والأكثر من ذلك فإن الشخص الذي يصدر الاستفسار لا يقلق بالنسبة لوجود البيانات في عقد مختلفة أو لكيفية تحقيق الاتصال. والعمل على مثل هذه النظم يعتبر في دور التجارب عند هذه النقطة.

وبعض العاملين في هذا المجال حاولوا أن يميزوا بين الحالات البسيطة والحالات الأكثر تعقيداً التي سبق ذكرها عن طريق الإشارة إلى الحالات الأولى بأنها بيانات منتشرة والحالة الثانية بأنها قواعد بيانات منتشرة. وبينما يمكن لمثل هذا التمييز أن يكون مفيداً إلا أنه من الصعب وضع أي من الأنواع المختلفة في سلة والآخرين في سلة أخرى وعلى هذا فلن نستمر في هذا التمييز هنا. وما سنفعله هو التعرض لموضوع قاعدة البيانات المنتشرة في صورة تقليدية traditional وذلك بوصف بعض الاختلافات الرئيسية لبدائل التصميم التي يجري العاملون في هذا المجال عليها التجارب.

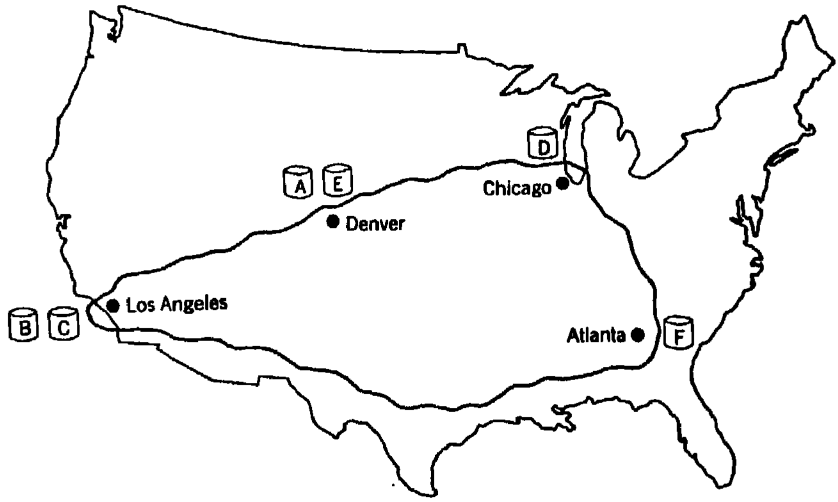
سوف ننظر، بصفة خاصة، إلى بعض خيارات التصميم لتخزين بيانات في الشبكة وبعض طرق معالجة مشكلة تجديد البيانات الموجودة في الشبكة إذا كانت مخزنة بصورة متكررة في أكثر من موقع واحد. وبعض الخيارات لتحقيق استفسار يتطلب بيانات مخزنة في موقعين أو أكثر في الشبكة وبعض الأفكار الخاصة بالكتالوجات التي تعطي معلومات عن أي البيانات مخزنة في أي العقد للنظام.

موقع البيانات:

معظم العمل على مفاهيم مطورة في قاعدة البيانات المنتشرة تم إعداده داخل محتوى المنهج العلاقي لإدارة قاعدة البيانات. وعلى هذا ففي مناقشة موقع

البيانات في هذا القسم وفي الأقسام التالية عن هذا الموضوع يكون هذا هو المنهج الذي نفترض استخدامه. وعلى هذا فسؤال موقع البيانات في شبكة منتشرة يصبح سؤالاً عن أخذ قرار بأي الملفات العـلاقية سـوف يوضـع عند أي عقـدة من عقد الشبكة. ومع اعتبارنا لكل امكانيات مواقـع البيانـات تذكـر أن الموضـوع الرئيسي يشمل ما إذا كان ممكناً تكرار الملفات في عقد مختلفة من عقد الشبكة أم لا.

وربما تكون أبسط طريقة لانتشار البيانات في الشبكة هو وضع ملفات محددة عند عقد محددة مع عدم حدوث ازدواج في البيانات. ويبين شكل 14.4 مثـل هذا الترتيب. والميزة هنا هي أنه عندما يجب تجديد بعض البيـانات، يحـدث التجديـد عند موقع واحد فقط وهو الذي يسهل من مراقبة سلامة البيانات.



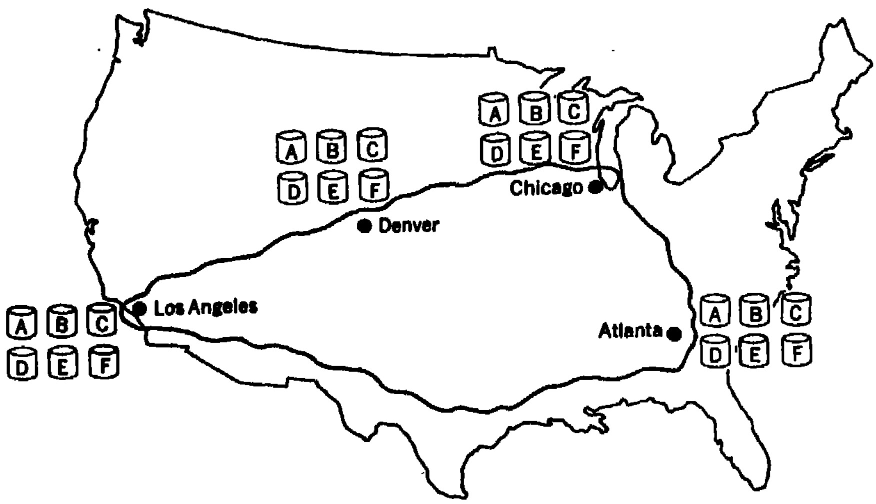
شكل 14.4

بيانات منتشرة بدون ازدواج

أحد العيوب هي أنه إذا حدث عـطل في إحدى النقـاط فلا يمكن الاتصـال بالبيانات الموجودة عند هذه النقطة بـواسطة أي شخص آخـر موجـود في الشبكة. وكذلك طبقاً لطبيعة البيانات والتطبيقـات قد يكـون هناك مصـاريف أكثر بـالنسبة للمال ووقت الاتصـال إذا احتـاجت العقــد المختلفـة بصفـة معتـادة إلى بيـانـات

تـوضع عنـد عقد إخـرى. ويمكن القول أن هـذا الترتيب يهـزم الكثير من أغـراض وجود نظام منتشر في المقام الأول.

ويأخذ الترتيب الموجود في شكل 14.5 الطرف الآخر فكـل الملفات مـزدوجة عند كل عقدة. والاتصال بالبيانات يكون سريع ورخيص بكل تـأكيد حيث أن كـل عمليـة استرجـاع بيانـات يمكن أداؤها عنـد العقدة التي تـطلبها تمـاماً. إلا أن كميـة المخازن المطلوبة والمشاكل المصاحبة بتنفيذ تجـديد البيـانات عنـد كل عقـدة من عقد الشبكة (حيث أن كل نقـطة لديهـا نسخة من كـل ملف) يجعل من هـذه الطريقـة طريقة غير واقعية لكل مواقف التطبيقات الفعلية.

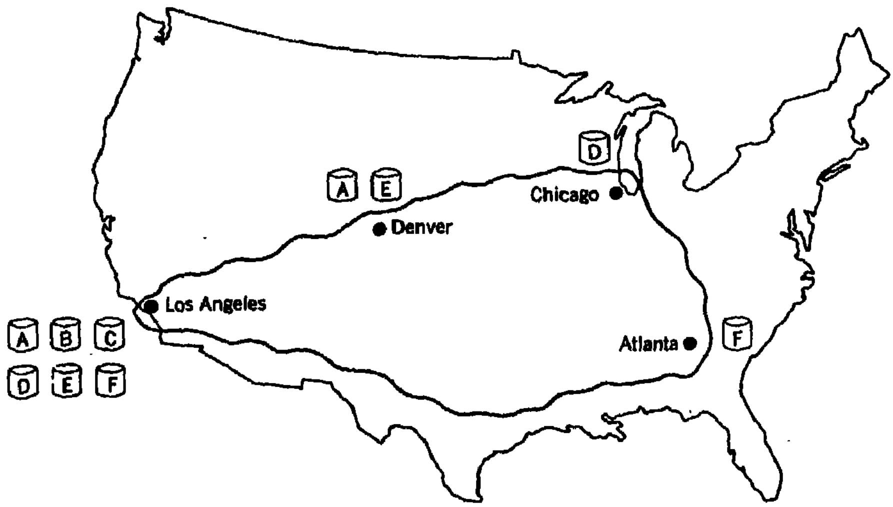


شكل 14.5

بيانات منتشرة مع وجود البيانات متكررة في كل موقع

ويبين شكل 14.6 صيغـة أكثر فـائدة لمفهـوم ازدواج البيانـات فكل الملفـات تخـزن عند إحـدى العقد (لـوس انجيلوس) بينما بعض الملفـات تخزن عنـد كـل من العقـد الأخرى. واختيـار موقـع تخزين الملفـات باستثنـاء الموقـع الذي يشمـل كـل الملفات يحدث طبقاً لسمة الإشـارة المحلية locality of reference. أي أنها تخـزن عند المواقع التي تتصل بها بصورة أكثر تكراراً وذلك يقلل من تكاليف الاتصـالات

التي تصاحب العقد غير المحلية في الشبكـة. أما إذا كـانت البيانـات المطلوبـة غير موجودة عند الموقع الذي يطلبها فلا يوجد هناك شـك عن أي العقد يجب اللجـوء إليها بالنسبة للبيانات. فدائماً يكون ذلـك عند العقـدة الموجـود بها نسخـة من كل الملفات. وحيث أنه يوجد ازدواج للبيـانات وبـالرغم من صغـر حجمه إلا أنـه ما يزال هناك موضوع تجديد بيانات مزدوجة عند أكثر من عقدة واحدة (عند عقدتـين بالضبط في هذه الحالة). وكذلك إذا ما تعطلت العقـدة الموجـود بها كـل البيانـات فيمكن أن يتعطل معظم عمـل الشبكة. مثـال ذلك إذا مـا احتاجت شيكـاغو إلى بيـانات مـوجودة في الملف وكـانت لوس انجيلوس معـطلة فلن تستطيـع شيكاغـو العمل.

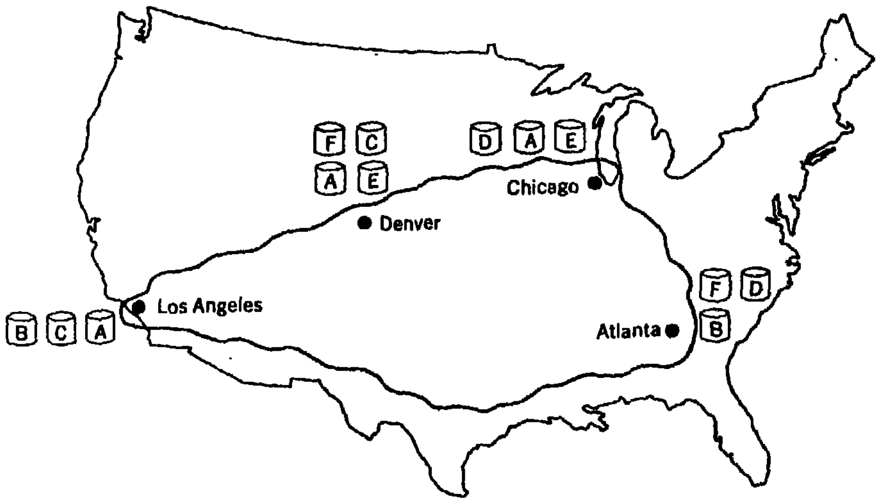


شكل 14.6

بيانات منتشرة مع وجود عقدة واحدة فقط لديها نسخة من كل البيانات

والترتيب الأكثر مرونة موضح في شكل 14.7. يمكن أن تزدوج الملفـات عند عدد من العقد التي يمكنها أن تكون ذات معنى بالنسبة لمتطلبات التطبيقات. وعـلى هذا فالملف الذي يستخدم الوقت كله تقريباً عند أحد العقد يخزن عند موقع هـذه العقدة فقط. أما الملف الذي يتكرر الإشـارة إليه بـواسطة الـبرامج أو المستفيـدين الموجودين في العديد من المواقع أو الذي يكون له متطلب حـرج بحيث أنه يكـون متاحاً عند بعض some المواقع طوال الـوقت (المشكلة هي إمكانيـة حدوث عـطل

بالمواقع) يمكن أن يحدث له ازدواج عند العديد من المواقع. ومرونة هـذا الترتيب تجعله أكثر جاذبية لبحث كل الامكانيات. وبينما يكون لهذا الترتيب مميزات ازدواج البيانات في النظام المنتشر إلا أنه يحتوي على مشاكل داخلية كذلك. هذه المشـاكل تشمل تجديد بيانـات مزدوجـة في مواقـع منفصلة وجمع بيـانات في مـواقع منفصلة مطلوبة في عملية وصل واحدة وإيجاد بيانات تخزن في مواقع قليلة فقط في الشبكة



شكل 14.7

بيانات منتشرة مع ازدواج بعض الملفات عند بعض العقد المختارة

وكنقـطة أخيرة في هـذا القسم لاحظ أننـا افـترضنـا أن سـواء عملنـا ازدواج أم لا للملفـات عند عقـد منفصلة من عقد الشبكـة فتظل كـل نسخـة من الملف أو الملف نفسه ككل. وهناك موضوع مرتبط كذلك بهذا الأمر وهو تجزئة الملفات إلى أجـزاء وتخزين الأجزاء المختلفة في عقد مختلفة. لقد اقترح الباحثـون تجزئـة الملفات أفقيـاً ووضع بعض سجلات الملف عنـد عقدة معينـة والبعض الآخر عنـد عقد أخـرى وتجزئتها رأسياً بوضع بعض حقول كـل (أو بعض) السجلات عنـد إحدى العقـد والحقول الأخرى لكل (أو لبعض) السجلات عند عقد أخرى. ومن الواضـح أن هذا الاقتراح أكثر تعقيداً.

السلامة : تجديد ومراقبة البيانات المزدوجة :

في آخر قسم ذكرنا عقدة وجود بيانـات مزدوجـة عند عقـد مختارة من الشبكـة المنتشرة يكـون لقاعـدة البيانـات المنتشرة ولاحـظنـا كـذلـك أن هـذا يقـدم بعض المشاكل. دعنا نلقي نظرة فاحصة أكثر عليها.

بملفات مزدوجـة عند مـواقع مختلفـة من الشبكة يصبـح تجديـد البيانـات في حـاجة إلى عمـل حيل لأدائـه. أنظر مـرة أخرى إلى الشبكـة المنتشرة الموجـودة في شكل 14.7. دعنا نقول أن المستفيد في أحد المواقع وليكن شيكاغـو يريـد أن يجدد قـطعة معينـة من البيانـات ولتكن الملف A. أول كـل شيء يجب أن يجـد النـظام نسخة للملف المطلوب فإذا كان الملف مخـزناً في مـوقع المستفيـد (كما هـو الحال في هذا المثال) فيكون هذا جيـد وإلا يجـب على النـظام أن يجد نسخـة من الملف في موقع آخر. (وبالتفكير الأكثر تقدماً في قاعدة البيانات المنتشرة تنقـل عملية تحـديد موقع البيانات إلى المستفيد). وفي أي حالة من الحالات يمكننـا: أن نتوقع أن يجدد النظام البيانات في نسخة الملف هذه (ملف A الموجود في شيكاغـو) ثم يجدد نفس البيـانات في النسـخ الأخرى للملف والمـوجودة في الشبكـة كلها (في دنفـر ولـوس انجيلوس). لكن ماذا يكون الحال عليه إذا كانت دنفر أو لـوس انجيلوس معطلة في هذا الوقت أو ماذا يحدث إذا تعطل النظام بعـد تجديـد بعض نسخ الملف دون إتمام بقيتها؟ النتيجة هي مشكلة في سلامة البيانات بوجـود قيم مختلفة لما يجب أن يكون له نفس البيانات في نسخ مختلفة من الملف في مواقع مختلفة توجـد فيها هـذه النسخ.

ويمكن وصف مشكلة مرتبطة بـذلك كـما يلي. في الفصـل 12 سبق أن ناقشنـا مشكلة التزامن التي تشمل مستفيدين يحاولان تجديد نفس قطعة البيـانات في نفس الوقت. وقد كان حل المشكلة في فصل 12 على الأقل محتويـاً في الحقيقة عـلى أن كل قـطعة بيـانات يـدور حولهـا الشك مـوجودة في ملف واحـد فقط من ملفات قـاعـدة البيانات. أما في حالة قاعدة البيانات المنتشرة ومع وجود ملفات مزدوجة في مـواقع مختلفـة يمكن أن يظل لـدينا حـالة المستفيـدان اللذان يحاولان تجـديد نفس قـطعـة البيانات في نفس الوقت. إلا أن أحد المستفيدين يمكن أن يتصل الآن بنسخـة من البيـانات في أحـد المواقـع لتجديـدها بينـما يتصل المستفيـد الآخر بنسخـة أخـرى

لتجديدها في موقع آخر. ومن الواضح أن هذا يعقد بشدة من المشكلة. فمثلاً ماذا يحدث إذا استخدمت قاعدة البيانات المنتشرة في تطبيق حجز مقاعد الطائرات. إفرض أنه لا يوجد أي مقعد خال في إحدى الرحلات ليوم معين وان المعلومات محددة بدقة في كل نسخة من نسخ الملف المناسب وليكن الملف A في شيكاغو ودنفر ولوس انجيلوس في شكل 14.7. من الممكن للبائع في موقعين وليكونا شيكاغو ودنفر أن يحاولا بيع مقعداً لعميلين مختلفين في نفس الوقت. فإذا قام النظام بتشغيل طلب شياغو مع نسخة ملف شيكاغو وطلب دنفر مع نسخة ملف دنفر فعند ذلك يبيع كل بائع نفس المقعد قبل أن توجد أي فرصة للنظام لتجديد كل النسخ الأخرى للملف والموجودة في النظام. ولن تكون نسخة الملف A في دنفر مهتمة بنسخة الملف A في شيكاغو والتي غيرت من عداد المقاعد من 1 إلى 0 والعكس صحيح. وعندما يقوم كل موقع بأخطار المواقع الأخرى بالتغيير يكون الوقت أصبح متأخراً جداً حيث أن المقعد قد تم بيعه مرتين.

دعنا نلقي نظرة على بعض الطرق المقترحة للتغلب على مشاكل السلامة هذه. إحدى الطرق مبنية على التوسع في مفهوم الإغلاق الذي سبق مناقشته في فصل 12. ولإنعاش هذه النقطة ففكرة الإغلاق لمستفيد واحد هي لجعله قادراً على التحكم على البيانات التي يقوم بتجديدها بحيث لا يستطيع أي شخص آخر الاتصال بها أو تعديلها (طبقاً للحالة نفسها) حتى يتركها هذا المستفيد. أما في حالة التجديد المنتشر فيحاول النظام أن يغلق كل نسخ الملف الموجودة في كل المواقع الموجودة بها قبل السماح بتجديد البيانات في أي نسخة من نسخ الملف.

وإحدى طرق تنفيذ هذا التجديد متعدد المواقع هو بسلسلة من العمليات التي تسمى بالإغلاق الشامل global blocking فيجب ألا تصدر الأوامر بالإغلاق للمواقع المختلفة بل يجب أن تستجيب هذه المواقع للإغلاق كذلك.

لاحظ التسلسل التالي (مستمراً في استخدام شكل 14.7 كمثال):

١ - أحد العقد ولتكن شياغو تصدر أمراً للعقد الأخرى الموجودة في الشبكة لإغلاق ملف وليكن الملف A. كما أنها تغلق كذلك نسختها من الملف A.

٢ - تغلق كل من دنفر ولوس انجيلوس، وكل منهما لديه نسخة من الملف A،

هذه النسخة (بمجرد أن يصبحا أحراراً من أي إغلاق آخر) ويقوما بأخطار شيكاغو.

٣ ـ بعد أن تعرف شيكاغو من دنفر ولوس انجيلوس إن النسخ قد أغلقت (مع ملاحظة أن نسختها هي الأخرى مغلقة) فإنها تصدر الأمر لتنفيذ التجديد على كل نسخ الملف A.

٤ ـ تجدد شيكاغو نسختها من الملف A وتفعل كل من دنفر ولوس انجيلوس نفس الشيء مع أخطار شيكاغو بأنهما نفذا ذلك.

٥ ـ عند ذلك تزيل شيكاغو الإغلاق عن نسختها من الملف A وتخطر كل من دنفر ولوس انجيلوس لعمل نفس الشيء مع نسخ الملف A الموجود لديهما.

لسوء الحظ بالإضافة إلى وجود تكاليف اتصالات كبيرة تصاحب ذلك فيمكن أن ينتج عن هذه الاجراءات إغلاقات ميته كثيرة مع وجود عمليات جارية عند مواقع مختلفة فتنتظر ملفات قام بإغلاقها آخرون. ويقود ذلك إلى إجراءات لاكتشاف مثل هذه الإغلاقات الميتة ثم إجبار أحد العمليات الجارية بترك إغلاقها للسماح لعملية جارية أخرى بالعمل ثم يعاد بدء العملية الجارية التي أجبرت على ترك إغلاقها لإتمامها.

ويشمل اختلاف لهذه الطريقة توضيح إحدى العقد على أنها عقدة محتوية dominant لكل من الملفات. وكل تجديدات البيانات في ملف معين يجب أن ترسل إلى العقدة التي يوجد عندها النسخة المحتوية dominent لهذا الملف. وبهذا النوع من التحديد يمكن للعقدة بالنسخة المحتوية أن تمارس درجة تحكم أكثر قوة على تجديدات ملف معين. ولسوء الحظ يمكن للعقدة المحتوية أن تصبح عنق زجاجة للملف مرتفع التقلبية. وكذلك إذا تعطلت إحدى العقد فتوقف كل أنشطة التجديد على هذا الملف إلا إذا ذهب النظام إلى مستوى آخر من التعقيد مع المقدرة على نقل الاحتواء dominance للملف من عقدة لأخرى.

وهناك طريقة أخرى كذلك للتحكم في التحديدات في بيئة قاعدة البيانات المنتشرة وهي لا تشمل أي إغلاق بل تحتوي على اختام للوقت time stamps. والفكرة هي تسجيل وقت بدء كل عملية جارية في قاعدة البيانات وتسجيل وقت

آخـر تجديـد وآخر قـراءة لكل قـطعة بيـانات أو لكـل سجل. وعنـد ذلـك يمكن التحكم في مشكلة السـلامة عن طـريق إجبـار كـل شيء في النـظام بـأن يبقى في تتابع. دعنـا نقول عـلى سبيل المثـال أن هناك عمليتـين جاريتـين X وY كل منهـما تحاول تجديد أحد سجلات الملف A في شيكاغو. ويبين تتابع الـوقت والاجراءات التالي كيفية تقدم أحد بدائل اختام الوقت العديدة:

10:00 AM العملية X تصل إلى السجل وتجدده.

10:01 العملية Y تصل إلى السجل وتجدده.

10:02 العملية Y تجدد السجل.

10:03 العملية X تحاول تجديد السجـل إلا أن حمايـة خاتم الـوقت تمنعها. وتلغى العملية X ويعاد بدءها بخاتم وقت جديد. لماذا؟

كجـزء من تشغيلها الابتـدائي فإن العمليـة الجاريـة X تقرأ القيم القـديمـة (10:00AM) من السجل. ولكن مع الوقت فإنها تكون مستعدة لتجديد البيـانات 10:03 تكـون هذه البيـانات قـد أصبحت متقادمـة بسبب العملية الجـارية Y التي قامت بتجديد السجل 10:02.

وصلات متعددة المواقع:

في بيئة قاعدة البيانات العلاقية المنتشرة الفعلية قـد تتطلب العمليـة الجاريـة التي تشمل وصلاً لملفـات موجـودة في موقعـين مختلفين أو أكـثر من الشبكة. وهنـاك طرق عديدة يمكن لمثل هذا الوصـل أن ينفذ بهـا ويمكن للاختيـار المناسب لـوصل معين أن يتسبب في فرق كبير في كمية الوقت اللازمة للاتصالات في الشبكة.

ربمـا يكون أبسط اختيـار هو إرسـال كـل الملفـات إلى أحـد المـواقـع وأجـراء الوصل هنـاك. ويأخـذ هذا وقتـاً كبيراً، وبصفـة خاصـة للملفات الكبـيرة. ومن الواضح أن اختيار موقع الهدف يكون حرجاً حيث أن إرسال ملف صغير إلى موقع يوجد فيه ملف كبير يأخذ وقتاً أقل كثيراً عما لو حدث العكس.

الـطريقـة الـرئيسيـة الأخـرى هي إرسـال أجـزاء من الملفـات فقط إلى عقـد أخرى لإجراء التشغيل عليها. فإذا كان ذلك مبنياً عـلى الاستفسار فيمكن تحـديد

سجلات معينة فقط وربما حقول معينة فقط في هذه السجلات والتي تكون مشمولة في الاستفسار. هذه تكـون هي البيانـات الوحيـدة اللازم إرسـالها إلى عقـدة أخرى لإتمام الوصل. فإذا كان هناك ثلاثة ملفات تدخل في العملية وإثنان منهما موجودان في مـوقع واحـد فيمكن وصلهما في مـوقعهما وتـرسل السجـلات المناسبـة فقط من النتائج لإجراء تشغيل آخر عليها.

وأخيراً هناك صيغة أخرى لآخـر طريقـة وهي إرسال البيـانات المنـاسبة من السجـلات الموجـودة في ملف عند عقـدة واحـدة إلى عقـدة أخـرى واحـدة في نفس الـوقت في صيغة من نـوع الفحص وذلك بـدلاً من إرسال سجـلات عديـدة مـرة واحدة لعمل وصل إضافية آخر.

إدارة الدليل:

سبق أن أشرنـا في هـذا الكتـاب إلى مجمـوعـات التحكم كـوسيلة لـوصف البيانات لنظام قاعدة البيانات. تذكر أنه كان هناك مجموعات تحكم أو بصفة عامة دلائل directories أو كتالـوجات catalogs للنـظام لوصف البيـانات الـواقعية أو المناظر المنطقية للبيانات ومحددات الأمن وما إلى ذلك. ومن الطبيعي أن تكون مثل هـذه المعلومات مـطلوبة في بيئة قاعـدة البيـانـات المنتشرة مـع إضـافـة واحـدة: المعلومـات الخاصـة بموقـع كل من ملفـات البيانـات في الشبكة. وبعـد كل شيء في منـاقشة قـاعدة البيـانـات وحتى هـذه العقـدة افـترضنـا أنـه عنـدمـا تحتـاج إحـدى العقد بيانات موجودة عند عقدة أخرى لمقابلة احتياج أحـد أنواع العمليـات الجاريـة فإنها تعرف knows مكان وجود هذه البيانـات. إلا أن هذه المعـرفة يجب أن تـأتي من مكان معين وهذا هو الدليل directory.

وهذا يحضر لنا سؤالاً هاماً. أين يخـزن الدليـل بنفسه؟ وكـما هو الحـال مع كيفيـة توزيـع ملفات البيـانات بنفسهـا فهناك عـدد من الإمكانيـات بعضها بسيط نسبياً والآخر أكثر تعقيداً مع وجود العديد من نفس أنواع المميزات أو العيوب التي سبق مناقشتها بالفعل. مثـال ذلك يمكن تخـزين محتويـات الدليـل في موقـع واحد فقط. وهـذا يسهل من عمـل التجديـد له وتعـرف كل عقـدة المكان الـذي تقصده بـالضبط للحصول عـلى معلومات من الـدليل. إلا أن كميـة تكاليف الاتصـالات

المشمـولة مـع كل العمليـات الجـاريـة تتجـه إلى عقـدة واحـدة حيث أن معلومـات الـدليل مـوجودة هنـاك. فإذا تعـطلت العقدة المـوجود بهـا الدليـل فيتعـطل الـنظام كله. وحتى بعد عمل نقطة الدليل فإنها تصبح عنق زجاجـة مع عمليـات جاريـة متراكمة من الشبكة كلها لحاجتها إلى معلومات من الدليل.

الطرف الآخر من الطيف بالنسبة لمحتويات الدليل هو وجـوده في كل مـوقع من مواقع الشبكة. وفي واقع الأمر لا يكون هذا سيئاً مثل محاولـة تخزين محتـويات قاعدة البيانات عند كل نقطة وذلك لقلة البيانات الموجودة في الدليل. وبالرغم من ذلك فكمية الوقت والطاقة اللازمـين لتجديـد الدليـل عند كـل عقدة لكـل تغيير في الدليل يمكن أن يجعل من هذا البديل عبئاً ويرفضه.

هنـاك طريقـة أخرى وهي البـدء بالمقـدمة بـأن كل عقـدة يكون لـديها دليـل يصف كل البيانات الموجودة عند هذه العقدة. لكن ماذا يحدث عنـدما تحتاج عملية جارية في عقدة معينة إلى بيانات موجودة في عقـدة أخرى؟ فليست محتـويات الـدليل بمفردها غير موجودة عند عقدة العملية الجارية بـل لا يوجـد كذلـك أي دليل بمـوقع محتـويات الـدليل. ويفـترض أن العملية الجـاريـة سـوف تبحث في كـل عقـدة عن معلومات الدليل المطلوبة. وعلى أية حال يمكننا التقدم خطوة أخرى للأمام بـوضع نسخة إضافيـة لمحتويـات الدليـل عند أحـد المواقـع الأخرى. ويمكن لكـل عملية جارية أن تعرف أين تذهب للحصول على معلومات غير موجودة في عقدة تواجدها. لكن القـول عن عقـدة الـدليـل بـأنها تصبـح عنق زجـاجـة ويتـوقف الـنظام إذا تعـطلت عقدة الـدليل يصبـح عامـلاً مؤثراً مـرة أخرى في هـذه الحالـة. وعلى هـذا فالتوسع المنطقي التالي يكون بالسماح بنسخ للأدلة الموجـودة في كل عقـدة توضـع في مواقع اختيارية عديدة من الشبكة. ويجب أن تصبح مميـزات وعيوب هـذا الترتيب واضحـة الآن. فأحـد التحسينات بـالنسبة لأي حـالة من الحـالات المحليـة وغـير المحلية يكون للعقدة هو تسجيـل معلومات الـدليل التي تـوجد في إشـارة غير محليـة وحفظ هذه المعلومات وقتياً (مرحلياً). وفي العديد من مواقف التطبيقات يعتقـدان العقدة قد تحتاج إلى نفس معلومات الدليل مـرة أخرى بعـد فترة وجيـزة وإيجادهـا محلياً يسرع من حدوث الأشياء.

وكملاحظة نهائية تأكد أن اختيار هذه الطرق لإدارة الـدليل يجب أن يعتمـد

بشدة على التطبيقات نفسها وبيئة تشغيل البيانات ككل. فـالعمليات الجـارية التي تحتاج إلى بيانات محلية في معظم الأحيان تحتاج إلى تخزين دليل البيانـات المحلية في الموقع المحلي. فإذا لم يكن هناك إشارات كـافية لبيـانات غـير محلية فيكـون تخزين العديد من نسخ أدلة كل النقاط في الشبكـة كلها غـير منتجاً. ومن نـاحية أخـرى فإن النسبة المرتفعة للإشارات إلى بيانات غـير محلية تشجـع على ازدواج معلومـات الـدلائل. والكميـة الكبيرة من نشـاط تجديـد الدليـل تشجع عـلى تقليـل ازدواج معلومات الدليل. ويجب أن توزن كل هذه العوامل بعناية.

قواعد البيانات على أجهزة الكمبيوتر الشخصية:

Database on Personal Computers

لم يمر وقت طويل بعد ظهور أجهزة الكمبيوتر الشخصية حتى ظهرت الحاجة إلى إدارة قواعد البيانات على هذه الأجهزة. وبالطبع بينما تكون المفاهيم الأساسيـة لقـاعدة البيـانات كـما هي بغض النظر عن حجم جهـاز الكمبيوتـر إلا أن طبيعـة الكمبيوتر الشخصي ونظم مكوناته تعطينا السبب لعمـل بعض التعليقات المحـددة هنـا عن قاعـدة البيانـات في بيئـة الكمبيـوتـر الشخصي. والنقـاط الأسـاسيـة عن الكمبيوتر الشخصي هي أنه عند هذه النقطة من الوقت تكون هذه الأجهزة بـطيئة في سرعة التشغيل ومحدودة جداً بالنسبة لسعة تخزينها للبيانـات كما أنها لـديها سعـة محـددة، أو لا يوجـد لديهـا سعة عـلى الإطلاق، للمشـاركة في البيـانات والـبرمجة المتعددة. وكذلك كما سوف نرى الحقيقة بأنها يمكن أن تتصل بأجهزة كمبيوتـر أكبر لها سعة تخزين أكبر هي حقيقة معروفة.

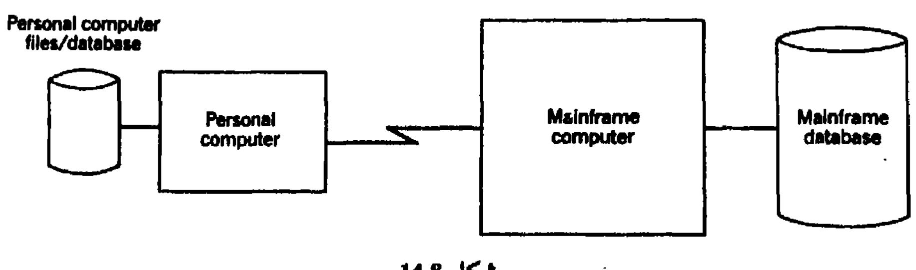
هنـاك عدد كبـير من منتجات نـظم برامـج إدارة قواعـد البيـانـات لأجهـزة الكمبيوتر الشخصية في السوق حالياً. وتختلـف هذه المنتجات بشدة في مقدرتها وفي أشكالها. ومعظمها، وليست كلها، مبنية على تكوين الملف الخطي البسيط. وهذا ليس مدهشاً وذلك بسبب المستفيدين غير التقنيين نسبيـاً لهذه الآلات. ومن ضمن السمات التي وجدت فيها توجد الفهارس والتأكد من مدى البيانـات في المدخـلات ومنتجات التقارير والرؤية المتعددة للبيانات.

وبعضها يتبع تعريف قاعدة البيانات المقدم في هذا الكتاب ويشمل إمكانيـة

الوصل بينها البعض الآخر لا ينطبق عليه ذلك. ويستخدم اصطلاح قاعدة البيانات بحرية أكثر على أجهزة الكمبيوتـر الشخصية عن استخـدامه مـع أجهزة الكمبيـوتر الأكبر.

العـديد من أجهـزة الكمبيوتـر الشخصية قـادرة عـلى الاتصـالات بـأجهـزة كمبيوتر كبيرة. وطبيعة العلاقة بين الاثنين مع ارتباطها بالبيانات وقـاعدة البيـانات تختلف. وربما أبسط شيء هو استخدام الكمبيوتـر الشخصي كنهايـة طرفيـة صماء. وفي هـذه الحالـة يمكن للكمبيوتـر الشخصي الاتصال ببيـانات مـوجودة في قـاعـدة بيانات الكمبيوتر الكبير مثلما تستطيع عمل ذلك أي نهاية طرفية أخرى مناسبـة إلا أنه لا يستطيع أن يسجل البيانات في أي صورة من الصور.

واستخـدام الكمبيوتـر الشخصي كنهـايـة طـرفيـة صـماء يمكن مقـارنتـه مـع استخدامه كنظام مستقل لتشغيل البيانات يتصل بنظام آخر أكبر منه كثيـراً لتشغيل البيانات أي بجهاز كمبيوتـر كبير. وأبسط صـورة في هذه الحـالة هي التي يحصـل الكمبيوتر الشخصي فيها على بيانات من الكمبيوتر الكبير ويرسل له بيانات كـذلك مـع عدم وجـود اشتباكـات خاصـة بين نـظم إدارة البيـانـات وقـاعـدة البيـانـات للجهـازين. مثال ذلـك يستطيـع الكمبيوتـر الشخصي إحضار بيـانات في ذاكـرتـه الرئيسية من قاعدة بيانات الكمبيوتر الكبير ويقوم بتشغيلها أو تخزينها بصورة معينة على أقراصه الخاصة به كما هو مبين في شكل 14.8 وفي الصيـغ الأكثر تـطوراً يمكن تشغيل نفس نظم برامج قاعدة البيانات في كل من الجهازين: الكمبيوتر الشخصي والكمبيوتر الكبير. ونظامي قاعدة البيانات هذين قد يعـرف أو لا يعرف كـل منهما الآخـر أو يعد كـل منهما للعمـل معاً وفي أي حـالة من الحـالات تكون البيـانـات مشكلة بطريقة مشتركة.



شكل 14.8

اتصال جهاز كمبيوتر كبير مع جهاز كمبيوتر شخصي

References مراجع

Bray, O.H., Distributed Database Management Systems, D.C. Heath, Lexington, MA, 1982.

Bray, O.H. and Freeman, H.A., Data Base Computers, D.C. Heath, Lexington, MA, 1979.

Champine, GA., «Six Approaches to Distributed Database», Datamation, vol. 23, no. 5, May, 1977.

Champine, G.A., «Four Approaches to a Data Base Computer», Datamation, vol. 24, no. 13, December, 1978, pp. 101 - 106.

Date, C.J., An Introduction to Database Systems, vol. 2, Addison - Wesley, Reading, MA, 1983.

Epstein, R., «Why Database Machines?», Datamation, vol. 29, no. 7, July, 1983, pp. 139 - 144.

IBM Corporation, C/CS/VS Version 1 Release 5 System/ Application Design Guide, IBM Form No. SC33-0068, 1980

IBM Corporation, IMS/VS Version 1 System Administration Guide, IBM Form No. SH20-9178, 1984.

Maryanski, F.J., «Backend Database Systems», ACM Computing Surveys, vol. 12, no. 1, March, 1980, pp. 3-26.

Mohan, C., «Recent and Future Trends in Distributed Data Base Management», Research Report RJ 4240, IBM Research Lab, San Jose, CA, May, 1984.

Rauzino, V.C., «The Present and Possible Future Data Base Machines», Computerworld, vol. 17, no. 23, June 6, 1983, pp. In Depth 1-20.

Rothnie, J.B., and Goodman, N., «A Survey of Research and Development in Distributed Database Management», Proceedings of the Third International Conference on Very Large Database, October 6-8, 1977, pp. 48-62.

أسئلة وتمارين: Questions and Exercises

١ ـ ما آلة قاعدة البيانات؟ ناقش اتجاهات مفهوم آلة قاعدة البيانات. هل تعتقد أنها مقنعة؟

٢ ـ ما مشغل نقطة النهاية التقليدي؟ كيف يمكن لمثل هـذه الوحـدة مساعـدة بيئة قاعدة البيانات؟

٣ ـ ما الذاكرة المصاحبة؟ كيف يمكن لمثل هذه التقنية أن تساعد بيئة قاعـدة البيانات؟

٤ ـ صف مميزات وعيوب منهج آلة قـاعدة البيـانات أي منهـا تعتقد أنـه مقبول أكثر؟ لماذا؟

٥ ـ ما قاعدة البيانات المنتشرة؟ وما نظام إدارة قاعدة البيانات المنتشرة؟

٦ ـ إعمل تخطيطاً للأوامر الخاصة بقاعدة البيانات المنتشرة.

٧ ـ صف الطرق المختلفة لتوزيع البيانات في إحدى الشبكات.

٨ ـ ما مميزات وعيوب الطرق المختلفة لتوزيع البيانات مع الشبكة؟

٩ ـ ما محلية الأدلة؟ ما درجة أهمية الدور الذي تلعبه في قاعدة البيانات المنتشرة؟ لماذا؟

١٠ ـما شعورك نحو تخزين بيانات مكررة في شبكـة منتشرة؟ وما مميـزات ذلك؟

١١ ـما هي مشاكل السلامة التي تصاحب ازدواج البيانات في شبكة منتشرة؟

١٢ ـصف اغـلاقات مـوت لقاعـدة بيـانـات منتشرة. مـا المشكلة التي تحـاول حلها؟ كيف يمكن تنفيذ الحل عملياً؟

١٣ ـصف اختـام الوقت لقـاعـدة بيـانـات منتشرة. مـا هي المشكلة التي تحـاول حلها؟ كيف يمكن تنفيذ الحل عملياً؟

١٤ ـمـا هي مميزات وعيـوب تجزئـة الملفات عـلى عقـد مختلفـة من عقـد الشبكـة بـالنسبة للوصـلات متعددة المـواقع؟ هـل هذا يستحق عمله أم أنـه مصـدر للمشاكل؟ لماذا ولم لا؟

١٥ ـنـاقش مشكلة إدارة الدليـل لقاعـدة بيـانـات منتشرة. هـل تعتقـد أن هـذا الموضوع أكثر حرجاً أو أقل حرجاً أو أنه متساوٍ في الحرج مع توزيع البيـانات نفسها؟ وضح إجابتك.

١٦ ـحدد الطرق التي يمكن استخـدام جهاز الكمبيـوتر الشخصي بهـا مع قـاعدة البيانات. ما مميزات وعيوب كل طريقة من هذه الطرق؟

الفصل الخامس عشر
المستقبل

THE FUTURE

النمو في قاعدة البيانات : The Growth of Database

والآن حصلت على الأوجه واسعة المدى لمجال قاعدة البيانات. ولن تستمر قواعد البيانات في كونها موضوعاً هاماً فقط في تشغيل البيانات بل إنها في الواقع سوف تزداد أهميتها مع زيادة فهم مزاياها ومع تطور العوامل العديدة التي تحد من استخدامها ليزداد هذا الاستخدام.

وبالنسبة لمميزات قاعدة البيانات فهي غير قابلة للرفض:

* إيجاد جو تعتبر فيه البيانات أحد موارد المؤسسة بصدق.
* تقليل تكرار البيانات للتجميع الذي يمكن إدارته لبيانات مرتفعة السلامة. فالتجميع المتكامل جيداً للبيانات يوفر مميزات مثل في دمج أنواع مختلفة للبيانات مع بعضها لإشباع احتياجات الاستفسارات المعقدة.
* المقدرة على معالجة العلاقات المتعددة بسهولة بين الكينونات المختلفة في بيئة العالم الواقعي.
* تشييد تجميع البيانات كمورد مستقل عن البرامج التي تستخدمها (استقلالية البيانات).
* حرية وجود مثل مواضيع الأمن والتزامن الاحتياطي والاستعادة والقابلية للمراجعة المحاسبية التي تدار بواسطة مجموعة من الاجراءات المعدة إعداداً جيداً والتي يوثق فيها.

وبعد ذلك هناك العوامل التي تحسن من التوسع في مميزات قـاعدة البيـانات من خلال تنفيذات مسئولة أكثر وبيئات أكثر كرماً :

* ستتزايد سرعة المشغلات ووحدات تخزين البيانات مع تقليل القيـود على الأداء تدريجياً والتي تحد من استخدام نظم إدارة قواعد البيانات في بعض الأحيان.

* ستقل تكاليف نظم المكونات مع إحضار مميزات قواعد البيانات لمستفيدين أكثر وأكثر. (بالطبع في وجهة نظر معينة في السنوات الأخيرة حدث ذلك بالفعل مع ظهور أجهزة الميكروكمبيوتر ونظم إدارة قواعد البيانات التي صممت لها).

* سيستمر تخفيف تعقيد نظم مكونات نظم إدارة قواعد البيانات.

* سيصبح منفذوا المؤسسات مؤيدين ويزداد تأييدهم لقواعد البيانات مع تمييزهم للطرق التي يمكن أن توسع بها قواعد البيانات إمكانياتهم لاتخاذ القرارات.

* ستقل تكلفة حفظ بيانات مكررة وبالتالي مشاكـل سلامـة البيانـات مع ازديـاد حفظ بيانات المؤسسة في نظم تشغيل بيانات آلية.

* ستزداد تكلفة صيـانة الـبرامج وهـذه المشكلة يمكن تقليلها بـواسطة استقـلالية البيانات ونمطيتها التي توفرها بيئة قاعدة البيانات.

* مجموعات الـبرامج والـبرامج النمـطية والتي يمكن أن تقلل من تكـاليف تطويـر التطبيقات سوف تزداد شعبيتها كما أنها سوف تتمتع بمزايا نفس نوع نمطيـة بيئة تشغيل البيانات التي تنتج من استخدام نظم إدارة قواعد البيانات.

* سوف يزداد إعداد أسطح بينية لقواعد البيانات صديقه للمستفيد تسمـح بمدى أوسع للمستفيدين غير التقنيين بالتمتع بمميزات البيانات المخزنة فيها.

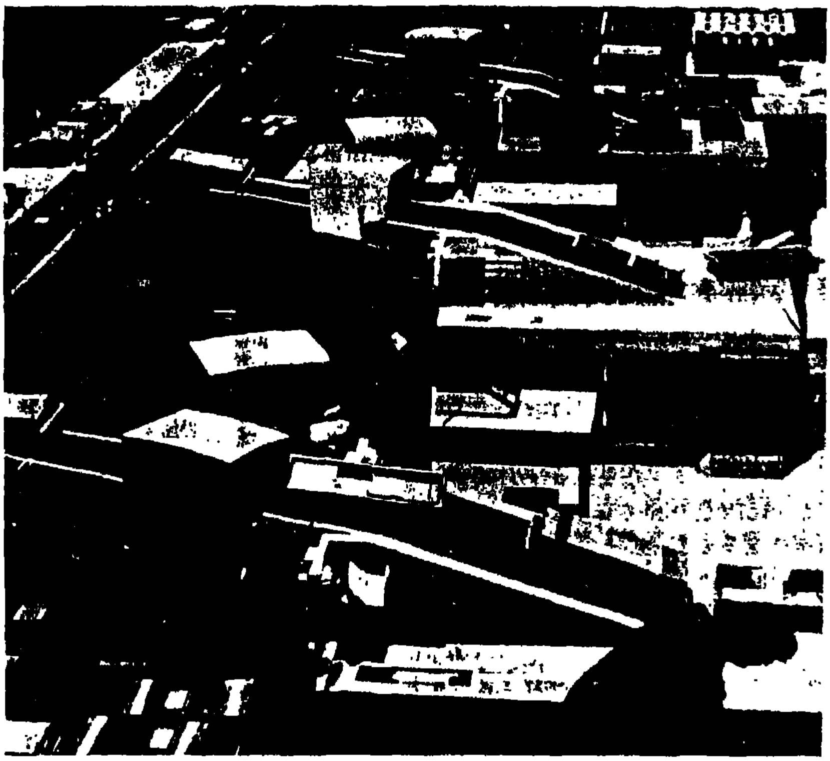
والعديد من هذه العوامل سوف يعطي فرصة لبعض التقنيات المعقـدة والتي سبق مناقشتها في هذا الكتاب أن تصبح حقائق تجـارية بـالفعل. فـأساليب نـظم المكونات ونظم البرامج المعدلة وبصفة خـاصة في مجـالات التزامن والاستعـادة مع قنوات الاتصالات الجـديدة والـرخيصة سـوف تساعـد على تـطوير مفهـوم قاعـدة البيانات المنتشرة. الاستمرار في التصغير الأكثر لنظم المكـونات مـع تقليل التكلفـة وازدياد السرعات قد يجعل من آلة قاعدة البيانات المصاحبة للذاكرة حقيقة واقعة.

مجالات تطبيقات مستقبلية : Future Application Areas

سبق أن تكلمنا عن العوامل التي تلاحظ في زيادة مستمرة لقـاعدة البيـانات في لقائها مع تطبيقات تشغيل البيانات «بالخبز والزبدة». وربما يكون تواجد قـاعدة البيانات كأساس للتوسع في مجالات التطبيقات الموجودة حاليـاً وتطويـر سلسلة من التطبيقات المدهشة أكثر إثارة.

فرسومات الكمبيوتر هي تطبيق مثير واقعي وقد حظي باهتمام واسع مؤخراً. ووعود الرسوم المتحركة الكمبيوتر ووسائل التصميم باستخدام الكمبيوتـر والعرض الصـوري لبيانـات الأعمال كـل هذه أمـور جيـدة وقـد بـدأ تمييـزهـا فقط. أحـد التحديات الرئيسية لقاعدة البيانات هي تطوير المقدرة على تخـزين بيانـات صوريـة في قواعد البيانات. فيجب أن تخزن البيانات بمثل هذه الطريقة التي يمكن أن تدير صوراً معقدة بطرق مفيدة ويمكن عرضها بسرعة على شاشات رسومـات ويمكن أن تقبـل تجديـدات سريعة لتعكس التغيـيرات التي أدخلها المستفيـدون عـلى الصـور المعروضة.

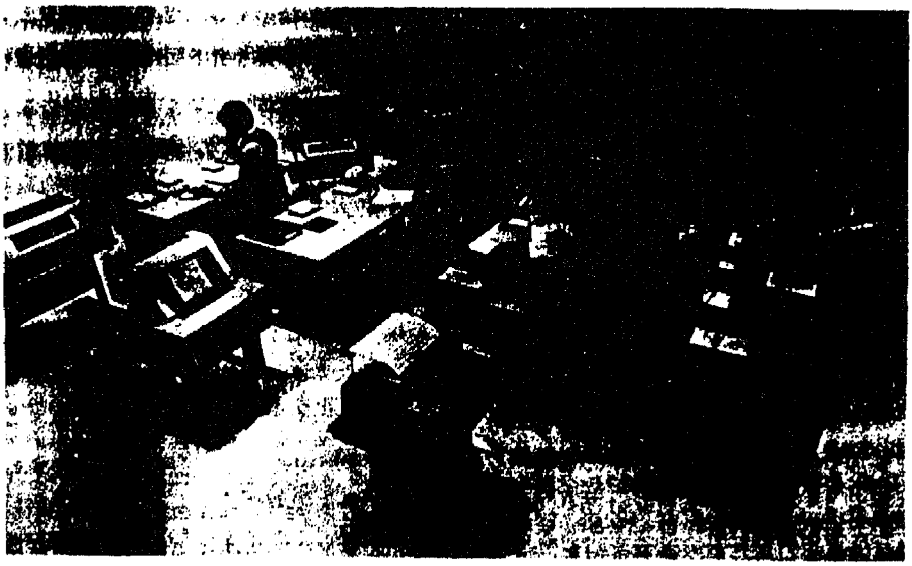
يجب أن تكون قاعدة البيانات قادرة على دعم مفهوم مصنع المستقبل والذي يشمل تكاملاً على مستوى مرتفع لآليات تحكم الكمبيوتر داخل عملية التصنيـع - شكـل 15.1. وفي هذا الصـدد يجب أن تكون قـاعدة البيـانات قـادرة على تخـزين بيانات رسومات تشمل تصميم عناصر يـراد تصنيعها ولبيـانات اختبـار لاختيارات المنتجات النهائية وبيانات لمراقبة وحدات الإنسان الآلي التي تقوم بالتصنيع وبيانات تستخلص حـالة عمليـات التصنيع المختلفـة وأدوات التصنيع المختلفـة وبيـانـات لتشغيـل البيانـات التقليدي لإدارة أجـزاء المخزون الموجودة في عمليـات التصنيع وتفاصيل الأوامر وتفاصيل الشحن وما إلى ذلك. والأكثر من ذلك يجب أن تعالـج كـل هذه البيـانـات بـطريقـة متكـاملة تكـامـلاً كبيـراً كـما أنها يمكن أن تقتسم في الاستعمال أو أنها تكون متداخلة مع بعضها بدرجة عالية أو كلاهما.



شكل 15.1

مصنع المستقبل (بإذن من شركة IBM)

وبتوجيه انتباهنا من بيئة أرضية المصنع إلى بيئة المكاتب فيجب أن تشكل قاعدة البيانات الأساس «لمكتب المستقبل». وهنا مرة أخرى يجب أن تكون قاعدة البيانات قادرة على إحداث ملاءمة وتكامل لمدى كبير من البيانات يشمل نصوصاً لتشغيل الكلمات وصوراً لتقديم الأعمال والبيانات الحرفية عددية المعتادة لكل أعمال حفظ السجلات المعتادة الموجودة في كل شيء من الأفراد الى المحاسبة في المكاتب التقليدية ـ شكل 15.2. وأحد أمثلة مثل هذا النظام هو «المكتب عن طريق المثال» وهو تتبع للاستفسار بواسطة الأمثلة (أنظر فصل 8). والذي يعالج ملفات وتقارير وما إلى ذلك وكلها كأشياء تقدم مستوى مرتفع جداً من المشاركة والتكامل في البيانات التي تدعمها كلها بكفاءة.



شكل 15.2

مكتب المستقبل (بإذن من شركة IBM)

وهنـاك مجال جـديد مثـير آخر معـروف بأنـه قاعـدة المعرفـة «Knowledge base» أو نظم الخبراء expert systems. وتغـذي مثل هـذه النظم بمعلومـات عن بعض مشـاكل العـالم الواقعيـة ويتوقـع منها بعـد ذلك من خـلال أساليب هـرمية وتداخلية، أن تستجيب لـلأسئلة الخاصـة بمجموعـة ظروف محـددة تخص المشكلة ومجـال موضـوعهـا. مثال ذلك، مثـل هـذه النـظم يمكن أن تحتـوي عـلى معـرفـة Knowledge عن مجال طبي معين وعند إعطائه مجموعة من الأعراض يتوقع النظام أن ينتـج تشخيصاً. ومن الـواضح أن قـاعدة المعـرفة المقصـودة ما هي إلا قـاعدة بيانات.

ويشمل تشغيل اللغات الطبيعية كمبيوتر قادر على تفسير وتحويل واستخـدام التعليمات والمعلومات المعطاة له في صورة لغة آدميـة. والقامـوس اللازم لمثـل هذا النشـاط وكذلـك عناصر التكـوين وقواعـد التفسير النحـوي يجب أن تخـزن كلهـا بغرض الاسترجاع مرتفع الكفاءة وهذه هي إحدى صور مشاكل قاعدة البيانات.

وهنـاك مجال آخـر لقاعـدة البيانـات وهو تخـزين واسترجـاع الصور وسـواء

استخدم تخزين واسترجاع بسيط أو تم تشغيله بسمات مميزة لها (حقل يسمى بتمييز الأنماط pattern recognition) فمثل هذه الصور تتطلب كما معتبراً من التخزين وتمثل نوعها الخاص من مشاكل قاعدة البيانات.

ولكن أبعد من المجالات المحددة التي ذكرت هنا فيقع التحدي الكبير في مشاركة أنواع مختلفة من التطبيقات في البيانات وفي تكامل البيانات المخزنة منفصلة (وربما مختلفة) لتطبيقات أخرى (كما سبق ذكر ذلك في زوج من تعليقاتنا). تخيل تصميم منتج آلياً متصل بالانتاج في مصنع آلي ومتصل بعمليات محاسبية مكتبية متصل بنظام استفسار خاص لمنفذي الإدارة العليا. تخيل أكثر هذه العمليات مستخدماً خليطاً من البيانات النصية والتقليدية والصور. وأخيراً تخيل أن كل هذه العمليات تعمل بسهولة ويسر من قاعدة بيانات واحدة متكاملة. هذا تحدي من تحديات المستقبل.

لقد قطعنا شوطاً طويلاً بالنسبة لقواعد البيانات في العشرين عاماً الماضية لكننا لم نفعل سوى أننا خدشنا السطح فقط. فسوف تصبح قواعد البيانات بؤرة تشغيل البيانات مع تمييزنا أكثر وأكثر أن اصطلاح تشغيل البيانات يعني أننا نهتم بمورد البيانات Data والذي يمكن أن يخدمنا بتشغيله بحكمه. هناك إمكانيات هائلة للنمو. فالطريق للأمام بالنسبة لقواعد البيانات سيكون مثيراً لمن يسير فيه.

References — مراجع:

Dertouzos, M.L., and Moses, J., eds., The Computer Age: A Twenty - Year View, MIT Press, Cambridge, MA, 1979.

Hayes - Roth, F., Waterman, D.A., and Lenat, D.B., eds., Building Expert Systems, Addison - Wesley, Reading, MA, 1983.

Sowa, J. F., Conceptual Structures, Addison - Wesley, Reading, MA, 1984.

Wegner, P., ed., Research Directions in Software Technology, MIT Press, Cambridge, MA, 1979.

Zloof, M.M., «Office - by - Example: A Business Language that Unifies Data and Word Processing and Electronic Mail», IBM Systems Journal, vol 21, no. 3, 1982, pp. 272 - 304.

أسئلة وتمارين: Questions and Exercises

١ - بتخيـل من عندك صف الأنـواع المختلفة للبيـانات التي يكـون هناك حـاجة إليهـا في بيئـة مكتب المستقبـل. كيف يمكن أن تقتسم مثـل هـذه البيـانـات بواسطة التطبيقات المختلفة التي سوف تتواجد هناك؟

٢ - بتخيل من عندك صف الأنواع المختلفة من البيانات التي يكون هناك حـاجة إليهـا في بيئة مصنـع المستقبـل. كيف يمكن أن تقتسم مثـل هـذه البيـانـات بواسطة التطبيقات المختلفة التي سوف تتواجد هناك؟

٣ - صف بكلمات من عنـدك التقنيـات والتغيـيرات التي يمكن تـوقعهـا في مجـال استخدام الكمبيوتر في المستقبل التي تشجع على استخدام قواعد البيانات.

٤ - فكر في أحد التطبيقات المبني على الصور ورسومات الكمبيـوتر ووصف كيف يمكن اقتسام البيانات المستخدمة في وصف هذه الصور مع تـطبيقات أخـرى لا يوجد بها صور.

٥ - هل تعتقد أنه في المستقبل القريب ستصبح الشركة قادرة عـلى امتلاك قـاعدة بيانات وحيدة تحتوي على كل بياناتها؟ لماذا ولم لا؟

٦ - هـل تعتقد أن مـدير ادارة البيـانات سـوف يقدم تقـاريره إلى مـدير تشغيـل البيـانـات في المستقبـــل؟ أم أن العكس يكـون صحيحـاً؟ أم أن الاثنـين سيصبحان متساويان؟ دافع عن رأيك مبيناً مبرراتك.

رقم الإيداع ٣٦٧٣/٩٤

مطابع المكتب المصرى الحديث
MODERN EGYPTIAN PRESS
ت : ٢٢١١٠٧١ - ٢٢١١٠٧٢ - فاكس ٢٢١١٠٧٣